ORIGINAL

Paid $1

FILED 14

2010 APR 15 A 10:58

RICHARD W. WIEKING
CLERK, U.S. DISTRICT COURT
N.D. CAL. S.J.

ADR

MATTHEW D. POWERS (Bar No. 104795)
matthew.powers@weil.com
STEVEN S. CHERENSKY (Bar No. 168275)
steven.cherensky@weil.com
WEIL, GOTSHAL & MANGES LLP
201 Redwood Shores Parkway
Redwood Shores, CA 94065
Telephone: (650) 802-3000
Facsimile: (650) 802-3100

Attorneys for Plaintiff
Apple Inc.

E-filing

UNITED STATES DISTRICT COURT

NORTHERN DISTRICT OF CALIFORNIA

SAN JOSE DIVISION

APPLE INC., a California Corporation,

Plaintiff,

v.

EASTMAN KODAK COMPANY, a New Jersey Corporation,

Defendant.

CV 10-01609 JCS

Case No.

COMPLAINT FOR PATENT INFRINGEMENT

DEMAND FOR JURY TRIAL

CERTIFICATION OF INTERESTED ENTITIES OR PARTIES

Plaintiff Apple Inc. ("Apple") complains and alleges as follows against Defendant Eastman Kodak Company ("Kodak"):

THE PARTIES AND THE NATURE OF THIS ACTION

1. Apple is a California corporation having its principal place of business at 1 Infinite Loop, Cupertino, California 95014.

2. On information and belief, Kodak is a New Jersey corporation with its principal place of business at 343 State Street, Rochester, New York, 14650. On information and belief, Kodak designs, manufactures, markets, and sells digital imaging devices and related software, including in this District and elsewhere in the United States.

3. This action arises under the patent laws of the United States, Title 35 of the United States Code, and relates to U.S. Patent No. 6,031,964 and U.S. Patent No. RE38,911.

JURISDICTION

4. This Court has subject matter jurisdiction under 28 U.S.C. § 1338(a).

5. Personal jurisdiction of this Court over Kodak is proper because Kodak commits acts of infringement in violation of 35 U.S.C. § 271 and places infringing products into the stream of commerce, through an established distribution channel, with the knowledge and/or understanding that such products are sold in the State of California, including in this District. These acts cause injury to Apple within the District. Upon information and belief, Kodak derives substantial revenue from the sale of infringing products distributed within the District, and/or expects or should reasonably expect its actions to have consequences within the District, and derives substantial revenue from interstate and international commerce. In addition, Kodak has knowingly induced, and continues to knowingly induce, infringement within this State and within this District by contracting with others to market and sell infringing products with the knowledge and intent to facilitate infringing sales of the products by others within this District, by creating and/or disseminating user manuals for the products with like mind and intent, and by warranting the products sold by others within the District.

VENUE

6. Venue is proper within this District under 28 U.S.C. §§ 1391(b), (c), and 1400(b).

INTRADISTRICT ASSIGNMENT

7. Plaintiff Apple is located in the San Jose Division and, therefore, assignment to the San Jose Division is proper and appropriate. Further, as set forth in Civil Local Rules 3-2(b) and (c), this action may be filed in and assigned to the San Jose Division because this action includes

intellectual property claims arising under the patent laws of the United States, Title 35 of the United States Code, and, therefore, may be assigned on a district-wide basis.

THE PATENTS

8. Apple is the owner by assignment of U.S. Patent No. 6,031,964 ("the '964 patent"), entitled "System and Method for Using a Unified Memory Architecture to Implement a Digital Camera Device," a true and correct copy of which is attached hereto as Exhibit A. The '964 patent was duly and legally issued on February 29, 2000, to Eric C. Anderson.

9. Apple is the owner by assignment of U.S. Patent No. RE38,911 ("the RE '911 patent"), entitled "Modular Digital Image Processing via an Image Processing Chain with Modifiable Parameter Controls," a true and correct copy of which is attached hereto as Exhibit B. The RE '911 patent was duly and legally reissued on December 6, 2005, to Eric C. Anderson and Gary Chin.

FIRST CLAIM FOR RELIEF
(Infringement of the '964 Patent)

10. Apple incorporates and realleges paragraphs 1 through 9 of this Complaint.

11. Apple is informed and believes, and on that basis alleges, that Kodak has infringed and continues to infringe the '964 patent by using, selling, and/or offering to sell, within the United States, and/or by importing into the United States, products, including, but not limited to, the Kodak Z Series of Cameras, including but not limited to the Z915, Z950, Z1085 IS, and Z1485 IS; the Kodak M Series of Cameras, including but not limited to the M340, M341, M380, M381, M530, M550, M1033, and M1093 IS; the Kodak C Series of cameras, including but not limited to the C142, C180, C182, C190, and C913; the Kodak SLICE camera; and the Kodak

video cameras, including but not limited to the Zi6, Zi8, Zx1, and the PlaySport;[1] which embody and/or practice at least claims 1-3 and 5-9 of the '964 patent in violation of 35 U.S.C. § 271.

12. Apple is informed and believes, and on that basis alleges, that Kodak has induced, and continues to induce, others to infringe the '964 patent in violation of 35 U.S.C. § 271 by taking active steps to encourage and facilitate direct infringement by others of at least claims 1-3 and 5-9 of the '964 patent with knowledge of that infringement, such as, upon information and belief, by contracting for the distribution of the infringing digital imaging devices for infringing sale such as by retail sales outlets, by marketing the infringing digital imaging devices, by creating and/or distributing user manuals for the infringing digital imaging devices, and by supplying warranty coverage for the infringing digital imaging devices sold in this State and in this District.

13. Apple is informed and believes, and on that basis alleges, that Kodak has contributorily infringed the '964 patent in violation of 35 U.S.C. § 271, by selling within the United States, offering for sale within the United States, and/or importing components that embody a material part of the inventions described in at least claims 1-3 and 5-9 of the '964 patent, are known by Kodak to be specially made or specially adapted for use in infringement of at least claims 1-3 and 5-9 of the '964 patent, and are not staple articles or commodities suitable for substantial, non-infringing use, including certain mobile devices and non-staple constituent parts of those mobile devices.

14. Apple will have put Kodak on notice of the '964 patent and Kodak's infringement thereof by no later than upon service of this Complaint.

---

[1] This section contains merely a shorthand summary of products currently accused of infringement by Apple. These descriptions, and the examples given therein, are not intended to exclusively define or otherwise limit the categories of Accused Products. Apple expects that Kodak will introduce additional products in the future that will infringe the Asserted Patents. Apple may, if necessary, amend or modify the above descriptions as discovery progresses.

15. Apple has been irreparably harmed by Kodak's acts of infringement, and will continue to be harmed unless and until Kodak's acts of infringement are enjoined and restrained by order of this Court.

16. As a result of Kodak's acts of infringement, Apple has suffered and will continue to suffer substantial damages in an amount to be proven at trial.

17. This case is "exceptional" within the meaning of 35 U.S.C. § 285, and Apple is entitled to an award of attorneys' fees.

SECOND CLAIM FOR RELIEF
(Infringement of the RE '911 Patent)

18. Apple incorporates and realleges paragraphs 1 through 9 of this Complaint.

19. Apple is informed and believes, and on that basis alleges, that Kodak has infringed and continues to infringe the RE '911 patent by using, selling, and/or offering to sell, within the United States, and/or by importing into the United States, products, including, but not limited to, the Kodak Z Series of Cameras, including but not limited to the Z915, Z950, Z1085 IS, and Z1485 IS; the Kodak M Series of Cameras, including but not limited to the M340, M341, M380, M381, M530, M550, M1033, and M1093 IS; the Kodak C Series of cameras, including but not limited to the C142, C180, C182, C190, and C913; and the Kodak SLICE camera;[2] which embody and/or practice at least claims 15-22, 27, 30-32, and 38-39 of the RE '911 patent in violation of 35 U.S.C. § 271.

20. Apple is informed and believes, and on that basis alleges, that Kodak has induced, and continues to induce, others to infringe the RE '911 patent in violation of 35 U.S.C. § 271 by

[2] This section contains merely a shorthand summary of products currently accused of infringement by Apple. These descriptions, and the examples given therein, are not intended to exclusively define or otherwise limit the categories of Accused Products. Apple expects that Kodak will introduce additional products in the future that will infringe the Asserted Patents. Apple may, if necessary, amend or modify the above descriptions as discovery progresses.

taking active steps to encourage and facilitate direct infringement by others of at least claims 15-22, 27, 30-32, and 38-39 of the RE '911 patent with knowledge of that infringement, such as, upon information and belief, by contracting for the distribution of the infringing digital imaging devices for infringing sale such as by retail sales outlets, by marketing the infringing digital imaging devices, by creating and/or distributing user manuals for the infringing digital imaging devices, and by supplying warranty coverage for the infringing digital imaging devices sold in this State and in this District.

21. Apple is informed and believes, and on that basis alleges, that Kodak has contributorily infringed the RE '911 patent in violation of 35 U.S.C. § 271, by selling within the United States, offering for sale within the United States, and/or importing components that embody a material part of the inventions described in at least claims 15-22, 27, 30-32, and 38-39 of the RE '911 patent, are known by Kodak to be specially made or specially adapted for use in infringement of at least claims 15-22, 27, 30-32, and 38-39 of the RE '911 patent, and are not staple articles or commodities suitable for substantial, non-infringing use, including certain mobile devices and non-staple constituent parts of those mobile devices.

22. Apple will have put Kodak on notice of the RE '911 patent and Kodak's infringement thereof by no later than upon service of this Complaint.

23. Apple has been irreparably harmed by Kodak's acts of infringement, and will continue to be harmed unless and until Kodak's acts of infringement are enjoined and restrained by order of this Court.

24. As a result of Kodak's acts of infringement, Apple has suffered and will continue to suffer substantial damages in an amount to be proven at trial.

25. This case is "exceptional" within the meaning of 35 U.S.C. § 285, and Apple is entitled to an award of attorneys' fees.

PRAYER

WHEREFORE, Apple requests that the Court:

1. Adjudge that Kodak has infringed, induced others to infringe, and/or committed acts of contributory infringement with respect to the asserted claims of the '964 and RE '911 patents;

2. Permanently enjoin Kodak from further infringement of the '964 and RE '911 patents;

3. Award Apple compensatory damages;

4. Award Apple its costs and reasonable experts' fees and attorneys' fees; and

5. Award Apple such other relief as the Court deems just and proper.

DEMAND FOR JURY TRIAL

Apple demands trial by jury for all issues so triable under Federal Rule of Civil Procedure 38(b) and Civil Local Rule 3-6(a).

Dated: April 15, 2010

WEIL, GOTSHAL & MANGES LLP

By: Steven S. Cherensky
Steven S. Cherensky
Attorneys for Plaintiff
APPLE INC.

CERTIFICATION OF INTERESTED ENTITIES OR PARTIES

As required by Civil Local Rule 3-16, the undersigned certifies that as of this date, other than the named parties, there is no such interest to report.

Dated: April 15, 2010

WEIL, GOTSHAL & MANGES LLP

By: Steven S. Cherensky
Steven S. Cherensky
Attorneys for Plaintiff
APPLE INC.

Exhibit A

US006031964A

United States Patent [19]

Anderson

[11] Patent Number: 6,031,964

[45] Date of Patent: *Feb. 29, 2000

[54] **SYSTEM AND METHOD FOR USING A UNIFIED MEMORY ARCHITECTURE TO IMPLEMENT A DIGITAL CAMERA DEVICE**

[75] Inventor: **Eric C. Anderson**, San Jose, Calif.

[73] Assignee: **Apple Computer, Inc.**, Cupertino, Calif.

[*] Notice: This patent issued on a continued prosecution application filed under 37 CFR 1.53(d), and is subject to the twenty year patent term provisions of 35 U.S.C. 154(a)(2).

[21] Appl. No.: **08/666,241**

[22] Filed: **Jun. 20, 1996**

[51] **Int. Cl.**[7] **H04N 5/225**

[52] **U.S. Cl.** **386/117**; 348/231; 348/552; 348/372

[58] **Field of Search** 358/909.1, 906; 386/38, 117; 348/207, 231–233, 552, 372; 395/750.08, 750.01, 677; 711/147, 148; H04N 5/225, 5/76, 5/92

[56] **References Cited**

U.S. PATENT DOCUMENTS

4,736,224	4/1988	Watanabe et al.	355/41
4,739,409	4/1988	Baumeister	348/244
5,027,150	6/1991	Inoue et al.	354/484
5,057,924	10/1991	Yamada et al.	358/906
5,077,582	12/1991	Kravette et al.	355/206
5,164,831	11/1992	Kuchta et al.	358/909.1
5,231,511	7/1993	Kodama et al.	358/906
5,247,321	9/1993	Kazami	354/106
5,262,868	11/1993	Kaneko et al.	358/909.1
5,359,427	10/1994	Sato	358/909.1
5,475,428	12/1995	Hintz et al.	348/263
5,475,441	12/1995	Parulski et al.	348/552
5,477,264	12/1995	Sarbadhikari et al.	348/231
5,486,853	1/1996	Baxter et al.	358/909.1
5,493,335	2/1996	Parulski et al.	348/233
5,539,535	7/1996	Aizawa et al.	358/909.1
5,576,757	11/1996	Roberts et al.	358/906
5,631,701	5/1997	Miyake	358/909.1
5,640,202	6/1997	Kondo et al.	358/909.1
5,867,214	2/1999	Anderson et al.	348/231

OTHER PUBLICATIONS

Martyn Williams, Review–NEC PC–DC401 Digital Still Camera, AppleLink Newbytes, Mar. 15, 1996, pp. 1–3.

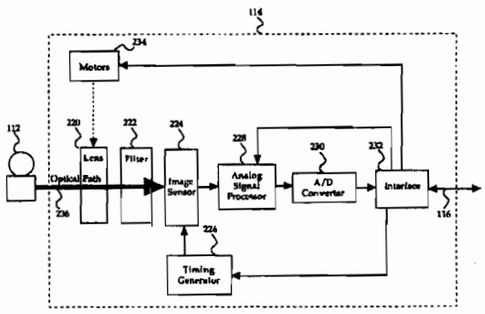
Primary Examiner—Thai Tran

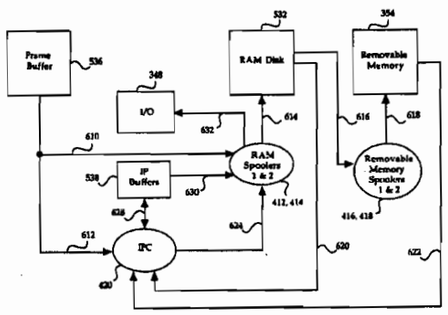
Attorney, Agent, or Firm—Carr & Ferrell LLP; Gregory J. Koerner

[57] **ABSTRACT**

A system and method for using a unified memory architecture to implement a digital camera device comprises a dynamic random-access memory for storing captured image data during processing and compression, a memory manager routine for allocating storage space within the dynamic random-access memory, a power management system for protecting the stored image data in case of a power failure, and an input/output interface for allowing an external host computer system to access the dynamic random-access memory.

16 Claims, 11 Drawing Sheets





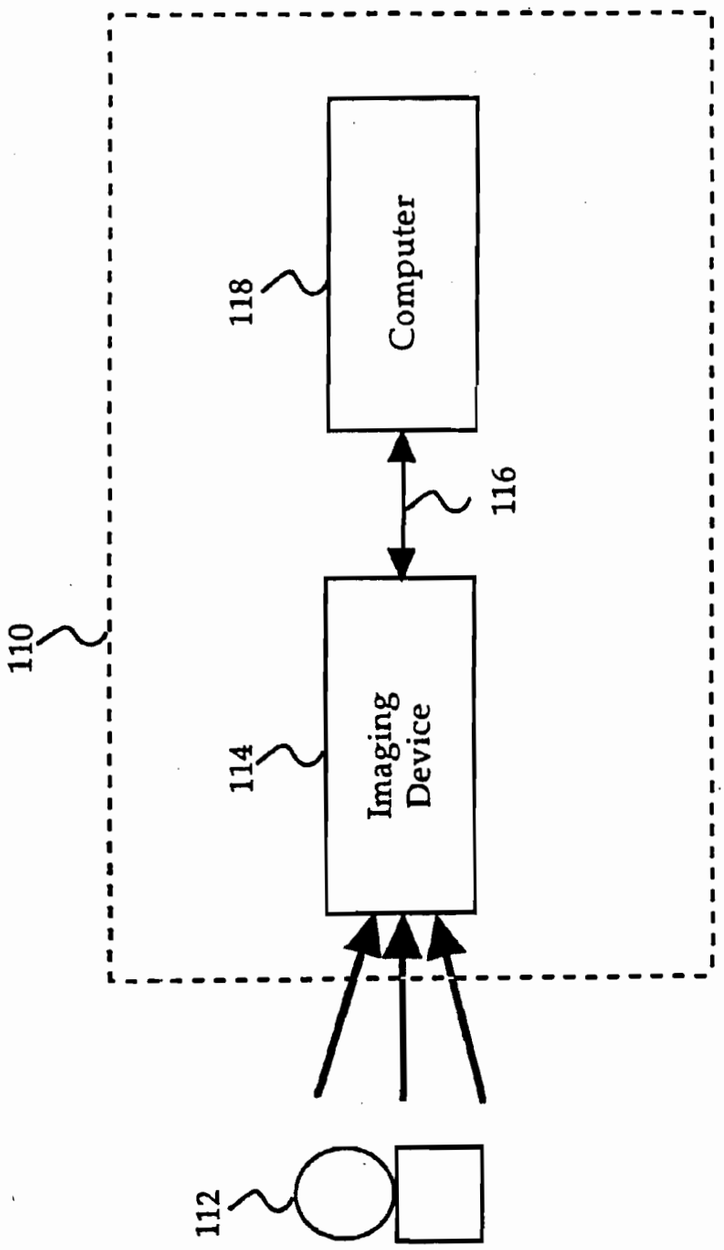


Fig. 1

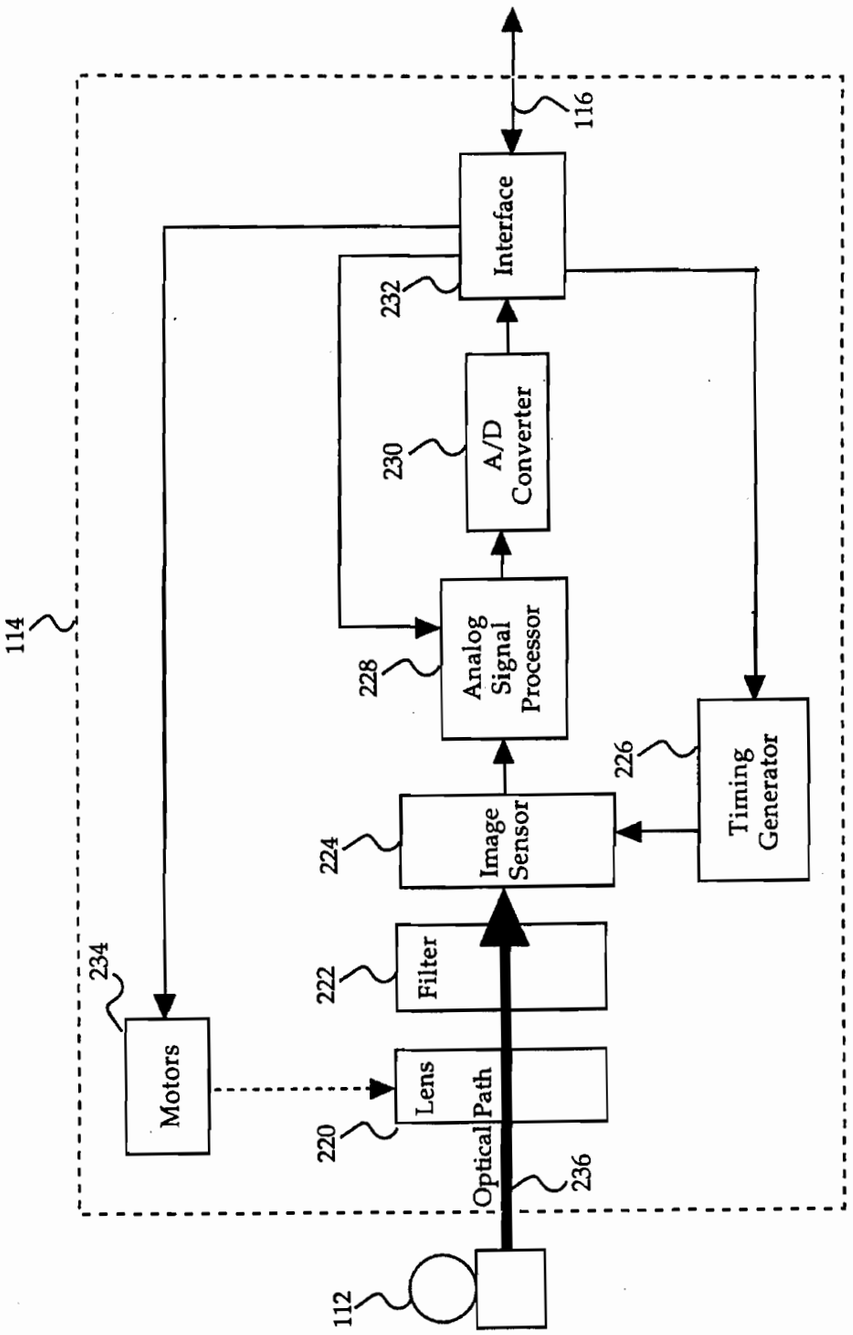

114
116
Interface
232
A/D Converter
230
Analog Signal Processor
228
Timing Generator
226
Image Sensor
224
Filter
222
Motors
234
Lens
220
Optical Path
236
112

Fig. 2

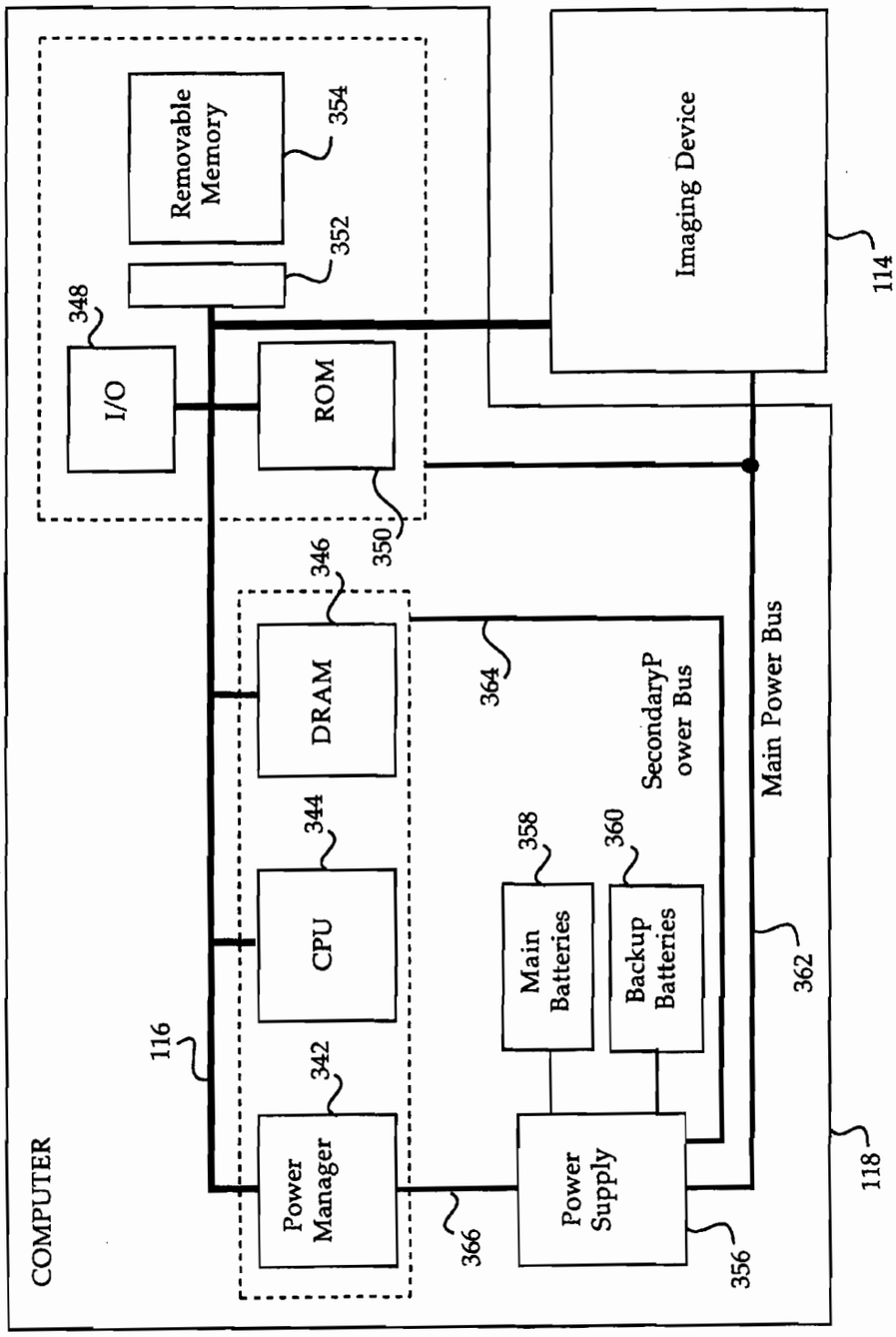

COMPUTER
116
118
Removable Memory
354
352
348
I/O
ROM
350
Imaging Device
114
346
DRAM
364
344
CPU
342
Power Manager
366
Main Batteries
358
Backup Batteries
360
Secondary Power Bus
Main Power Bus
362
Power Supply
356

FIG. 3

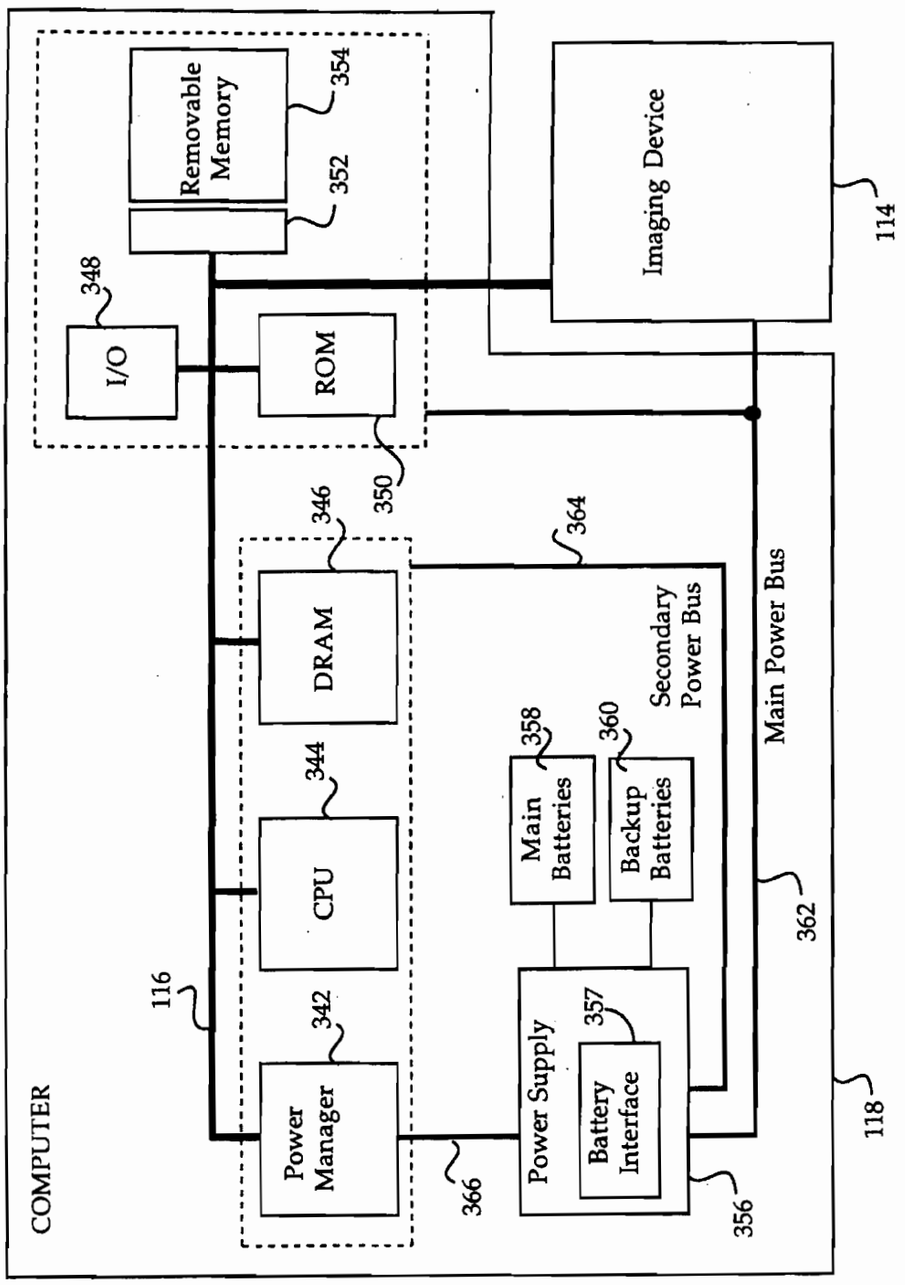

COMPUTER
Removable Memory
354
352
348
I/O
ROM
350
346
DRAM
364
344
CPU
116
342
Power Manager
366
Main Batteries
358
Backup Batteries
360
Secondary Power Bus
Power Supply
357
Battery Interface
356
Main Power Bus
362
118
Imaging Device
114

FIG. 4

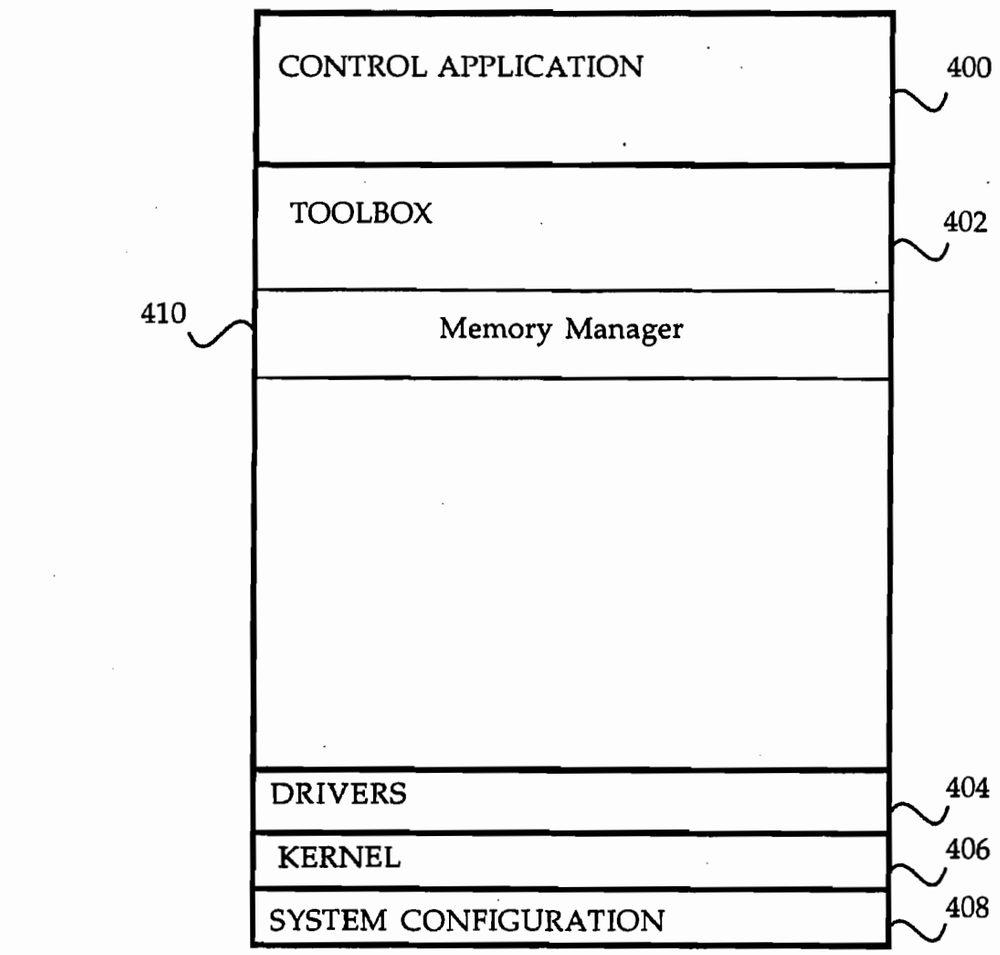

CONTROL APPLICATION
400
TOOLBOX
402
410
Memory Manager
DRIVERS
404
KERNEL
406
SYSTEM CONFIGURATION
408
350

FIG. 5

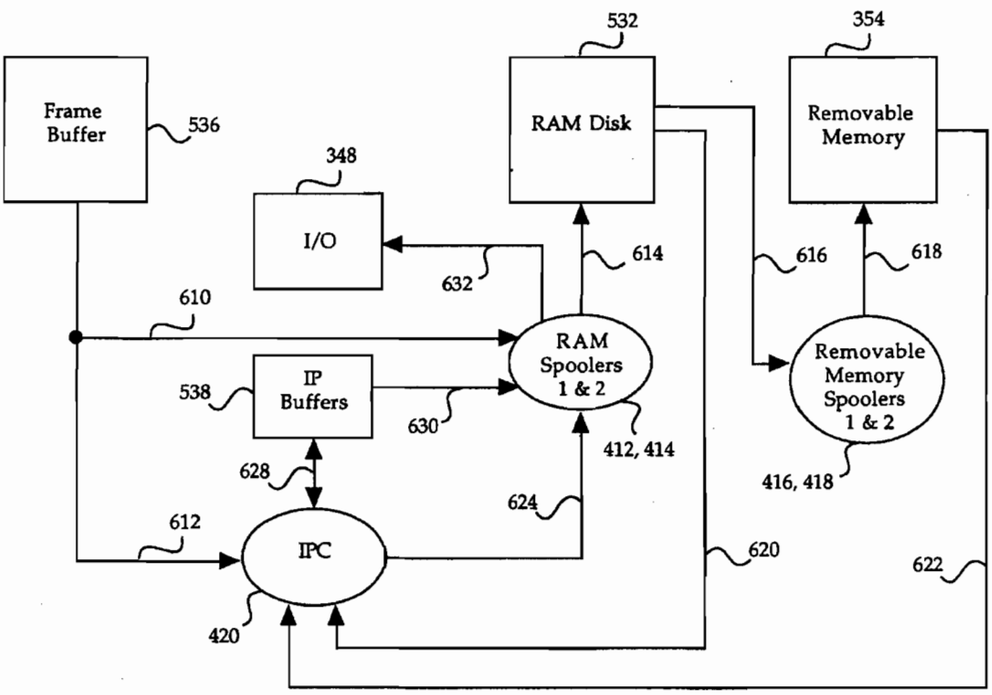

Frame Buffer
536
348
I/O
532
RAM Disk
354
Removable Memory
632
614
616
618
610
538
IP Buffers
RAM Spoolers 1 & 2
630
412, 414
Removable Memory Spoolers 1 & 2
416, 418
628
624
612
IPC
620
622
420

FIG. 6

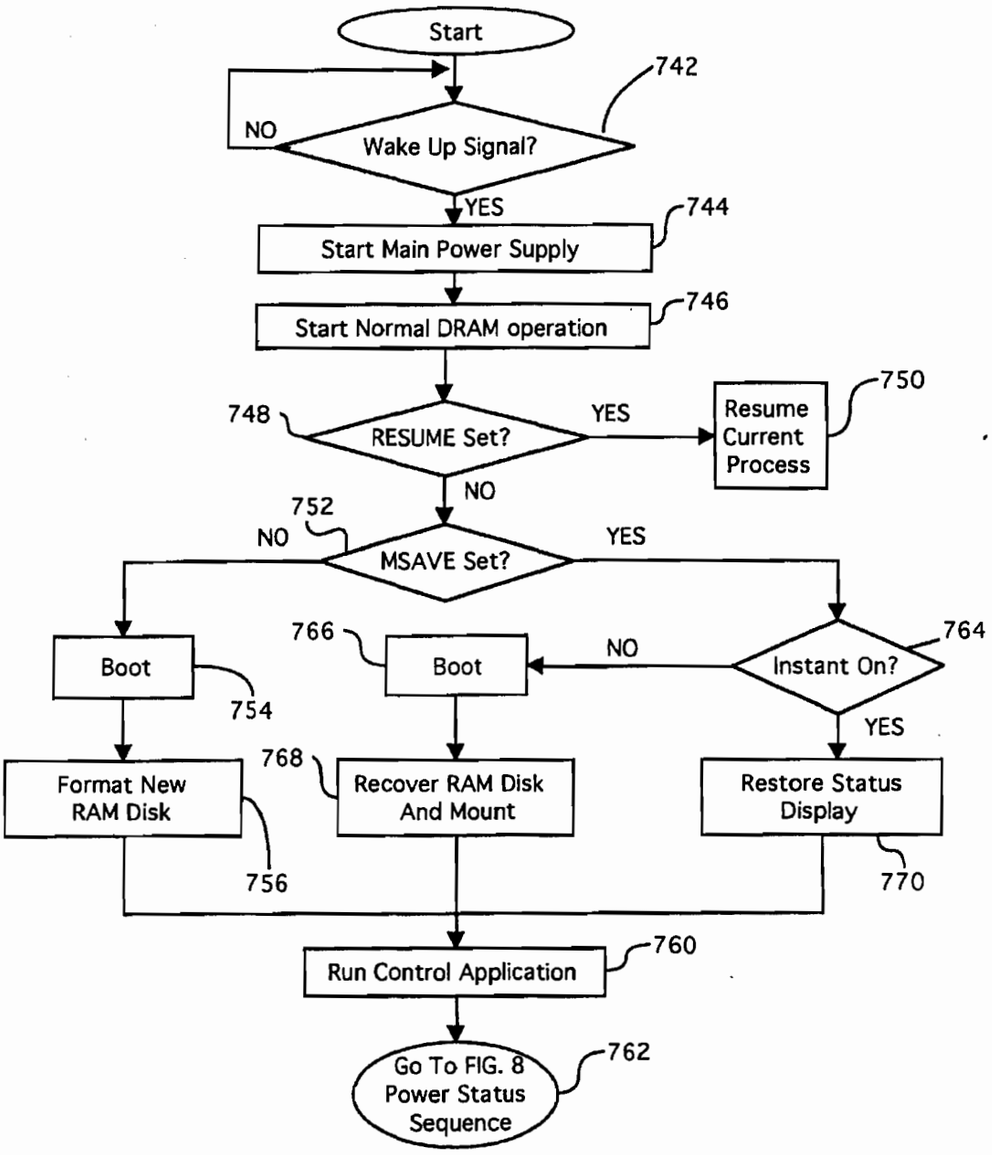

Start
742
NO
Wake Up Signal?
YES
744
Start Main Power Supply
746
Start Normal DRAM operation
748
RESUME Set?
YES
750
Resume Current Process
NO
752
NO
MSAVE Set?
YES
766
Boot
NO
764
Instant On?
Boot
754
YES
768
Format New RAM Disk
Recover RAM Disk And Mount
Restore Status Display
756
770
760
Run Control Application
762
Go To FIG. 8 Power Status Sequence

FIG. 7

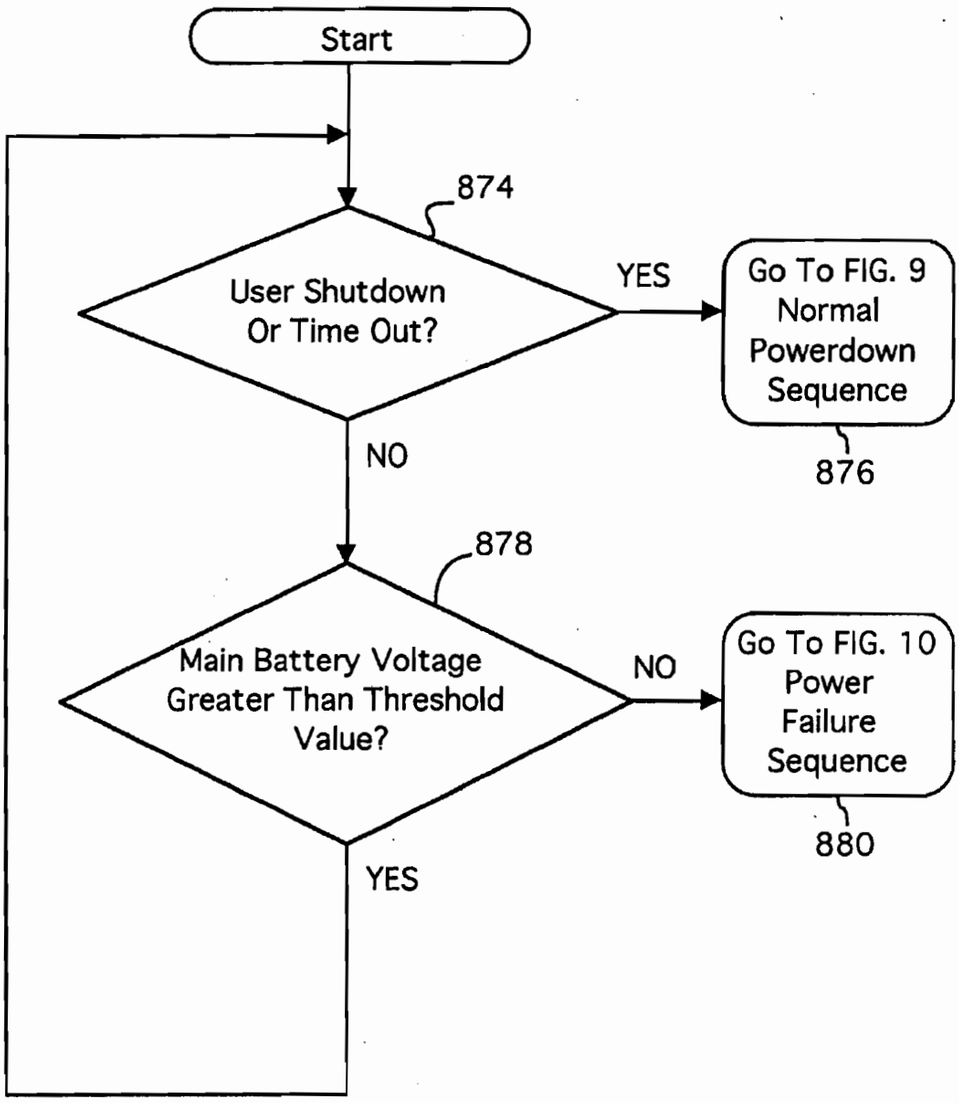

Start
874
User Shutdown Or Time Out?
YES
Go To FIG. 9 Normal Powerdown Sequence
876
NO
878
Main Battery Voltage Greater Than Threshold Value?
NO
Go To FIG. 10 Power Failure Sequence
880
YES

FIG. 8

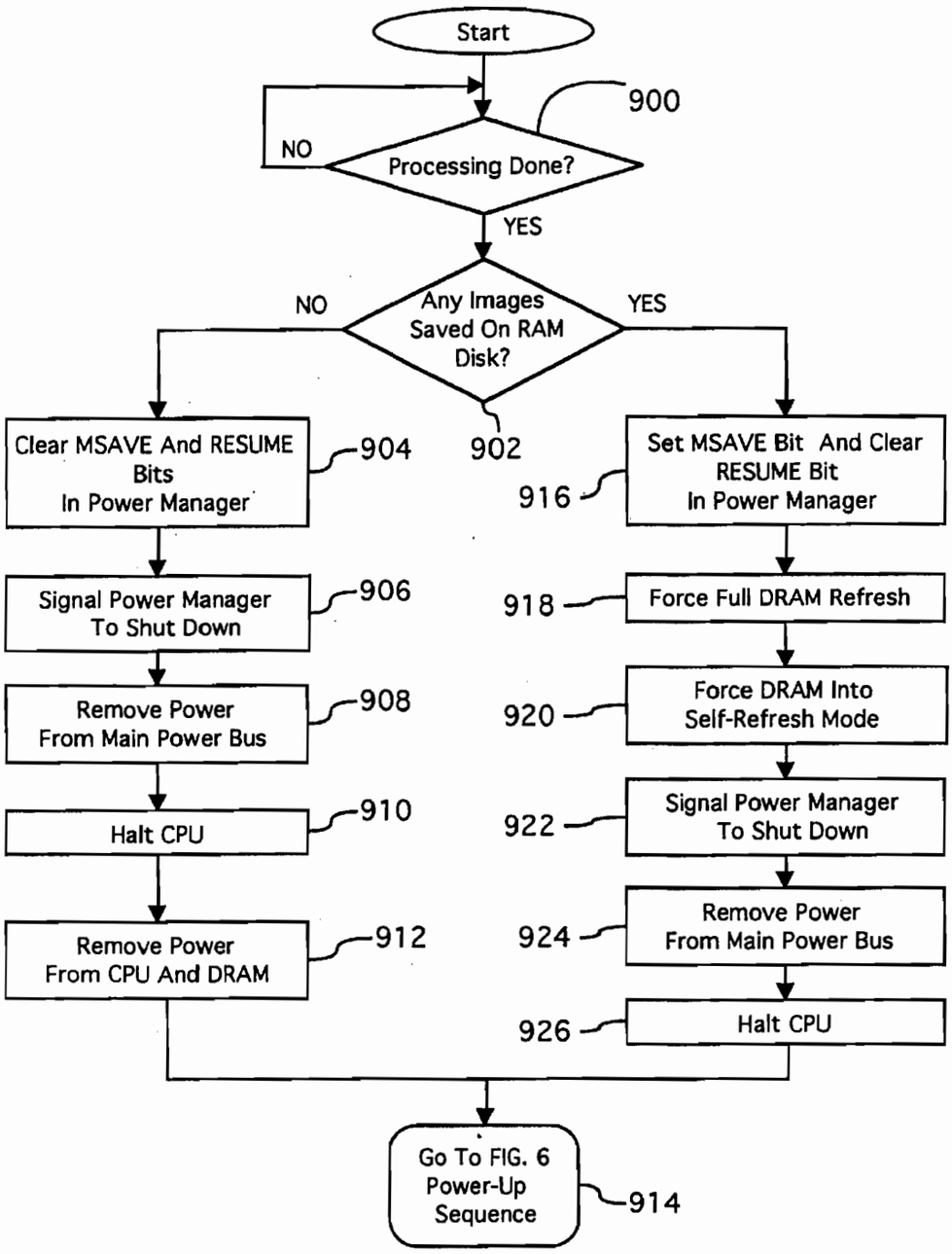

Start
900
Processing Done?
NO
YES
Any Images Saved On RAM Disk?
NO
YES
902
Clear MSAVE And RESUME Bits In Power Manager
904
Signal Power Manager To Shut Down
906
Remove Power From Main Power Bus
908
Halt CPU
910
Remove Power From CPU And DRAM
912
Set MSAVE Bit And Clear RESUME Bit In Power Manager
916
Force Full DRAM Refresh
918
Force DRAM Into Self-Refresh Mode
920
Signal Power Manager To Shut Down
922
Remove Power From Main Power Bus
924
Halt CPU
926
Go To FIG. 6 Power-Up Sequence
914

FIG. 9

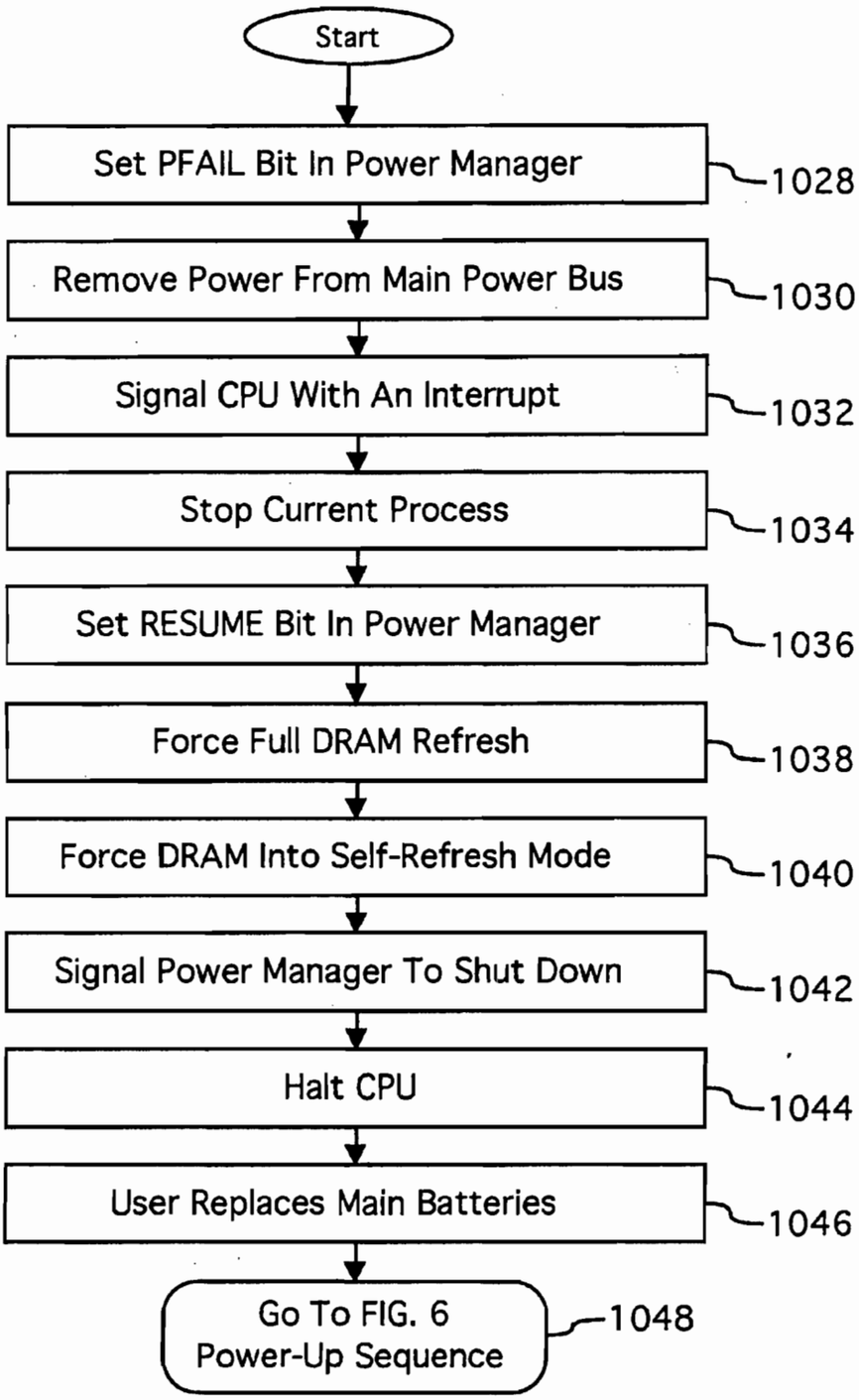

Start
Set PFAIL Bit In Power Manager
1028
Remove Power From Main Power Bus
1030
Signal CPU With An Interrupt
1032
Stop Current Process
1034
Set RESUME Bit In Power Manager
1036
Force Full DRAM Refresh
1038
Force DRAM Into Self-Refresh Mode
1040
Signal Power Manager To Shut Down
1042
Halt CPU
1044
User Replaces Main Batteries
1046
Go To FIG. 6
Power-Up Sequence
1048

FIG. 10

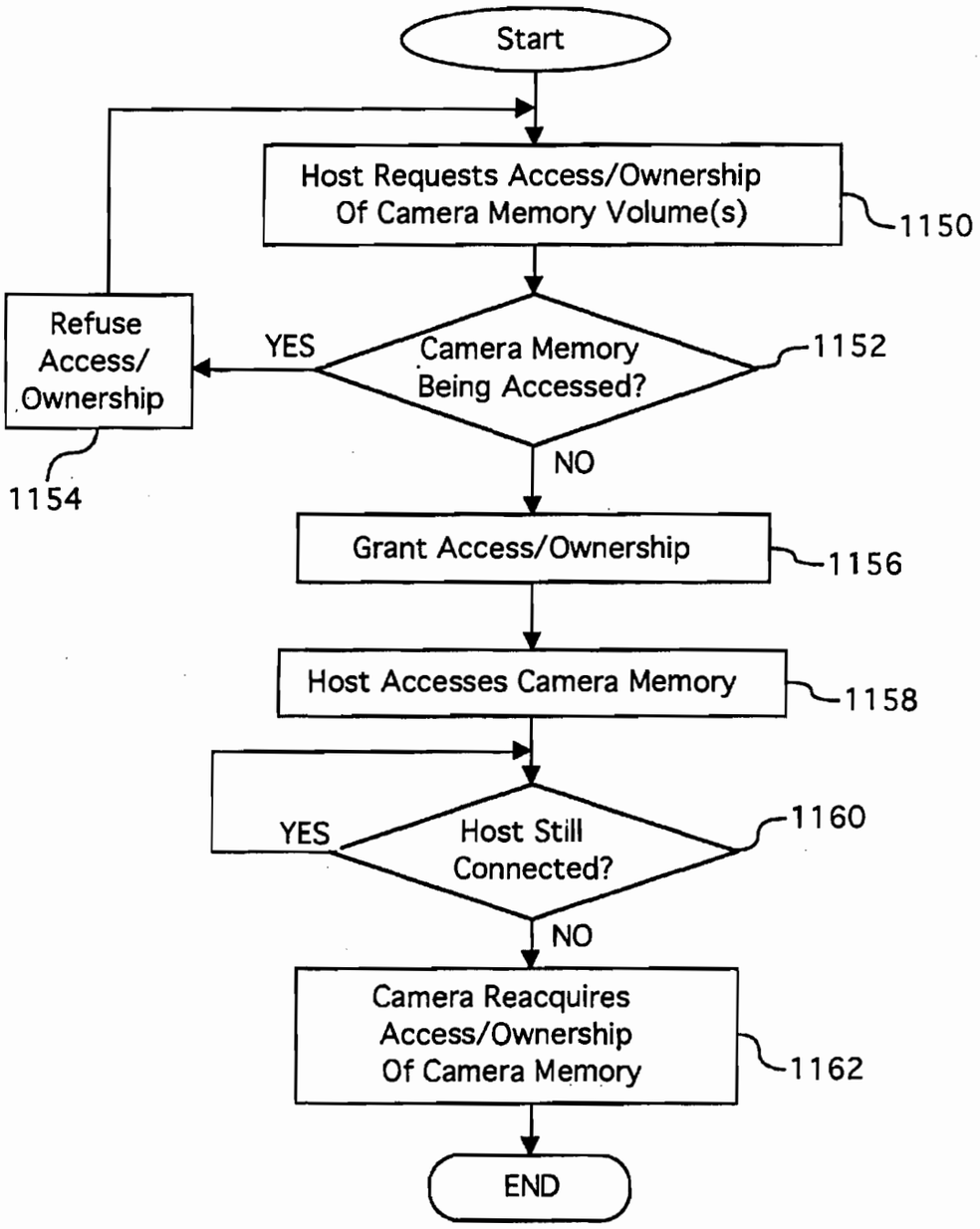

Start
Host Requests Access/Ownership Of Camera Memory Volume(s)
1150
Camera Memory Being Accessed?
1152
YES
Refuse Access/ Ownership
1154
NO
Grant Access/Ownership
1156
Host Accesses Camera Memory
1158
Host Still Connected?
1160
YES
NO
Camera Reacquires Access/Ownership Of Camera Memory
1162
END

FIG. 11

SYSTEM AND METHOD FOR USING A UNIFIED MEMORY ARCHITECTURE TO IMPLEMENT A DIGITAL CAMERA DEVICE

CROSS-REFERENCE TO RELATED APPLICATIONS

This application relates to co-pending U.S. patent application Ser. No. 08/631,173, entitled "Apparatus And Method For Increasing A Digital Camera Image Capture Rate By Delaying Image Processing," filed on Apr. 11, 1996, and to co-pending U.S. patent application Ser. No. 08/633,105, entitled "A System And Method For Managing Utilization Of A Battery," filed on Apr. 16, 1996, which are hereby incorporated by reference.

BACKGROUND OF THE INVENTION

1. Field of the Invention

This invention relates generally to digital cameras and more particularly to a system and method for using a unified memory architecture to implement a digital camera device.

2. Description of the Background Art

Modern digital cameras typically include an imaging device which is controlled by a built-in computer system. The built-in computer system accesses raw image data captured by the imaging device and then processes and compresses the data before storing the compressed data into an internal memory. Efficient operation of the built-in computer is therefore an important consideration for camera designers and manufacturers. The memory architecture of a particular computer system determines data storage techniques and can thus significantly effect the operational efficiency of the entire digital camera system.

In conventional hardware-based digital cameras, the system memory architecture is typically implemented using multiple discrete blocks of memory. The conventional digital camera captures raw image data and then remains unusable until the data is completely processed and stored into internal flash memory. This conventional memory architecture does not readily support the rapid capture of a series of image data sets and is therefore somewhat inconvenient in certain photographic applications.

Furthermore, in modern computer systems, a multi-threading environment effectively allows multiple system processes to run concurrently. The multi-threading environment may thus permit a digital camera to more efficiently process, compress and store image data by performing these functions in the background, when the built-in computer is not busy with more important tasks such as capturing additional sets of image data. A memory architecture which supports the background spooling functions can therefore significantly increase the computer system efficiency. Therefore, an improved system and method is needed for using a unified memory architecture to implement a digital camera device.

SUMMARY OF THE INVENTION

In accordance with the present invention, a system and method are disclosed for using a unified memory architecture to implement a digital camera device. In the preferred embodiment of the present invention, the digital camera includes an imaging device for capturing raw image data and a computer for processing, compressing and storing the image data. The computer includes a dynamic random-access memory (DRAM), a central processing unit (CPU), a memory manager routine stored in read-only memory (ROM), a power management system, an input/output interface (I/O), and an optional removable memory.

The DRAM includes frame buffers for storing raw image data received from the imaging device, a working memory area, and a RAM disk with a standardized file system. In the preferred embodiment, the CPU executes a memory manager routine which selectively allocates storage locations within the DRAM, depending on the requirements of the system and the current image data. The power management system includes a power manager which controls a power supply that is powered by main batteries and which is also connected to backup batteries, in case of a power failure in the main batteries. If the main batteries fail to provide sufficient power for operating the entire digital camera, the power manager maintains operating power to itself, the CPU and the DRAM using the backup batteries.

The DRAM uses a standardized file system which enables an external host computer system to readily access image data stored in the DRAM via the digital camera's I/O interface. The nonvolatile removable memory serves as a storage area for additional image data. A camera user who possesses several removable memories may thus replace a full removable memory with an empty removable memory to effectively expand the picture-taking capacity of the digital camera. The present invention therefore improves the performance and efficiency of a digital camera device through the use of an improved unified memory architecture.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. **1** is a block diagram of a digital camera according to the present invention;

FIG. **2** is a block diagram of the preferred embodiment for the imaging device of FIG. **1**;

FIG. **3** is a block diagram of the preferred embodiment for the computer of FIG. **1**;

FIG. **4** is a memory map showing the preferred embodiment of the Read-Only-Memory (ROM) of FIG. **3**;

FIG. **5** is a memory map showing the preferred embodiment of the Dynamic Random-Access-Memory (DRAM) of FIG. **3**;

FIG. **6** is a block diagram showing preferred data paths for transmitting image data between components of the FIG. **3** computer;

FIG. **7** is a flowchart of preferred method steps for performing a power-up sequence according to the present invention;

FIG. **8** is a flowchart of preferred method steps for performing a power status sequence according to the present invention;

FIG. **9** is a flowchart of preferred method steps for performing a normal power-down sequence according to the present invention;

FIG. **10** is a flowchart of preferred method steps for performing a power failure sequence according to the present invention; and

FIG. **11** is a flowchart of preferred method steps for a host computer system to access the camera memory of FIG. **3**.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENT

The present invention discloses a system and method for using a unified memory architecture to implement a digital camera device and comprises a unified memory device for storing sets of image data and a central processing unit for

executing a memory manager routine which allocates storage locations within the memory device to store the sets of image data. The present invention also includes a power management system for protecting the sets of image data in the event of a power failure to the digital camera device and an input/output interface whereby an external host computer can access the sets of image data stored in the memory device.

Referring now to FIG. **1**, a block diagram of a camera **110** is shown according to the present invention. Camera **110** preferably comprises an imaging device **114**, a system bus **116** and a computer **118**. Imaging device **114** is optically coupled to an object **112** and electrically coupled via system bus **116** to computer **118**. Once a photographer has focused imaging device **114** on object **112** and, using a capture button or some other means, instructed camera **110** to capture an image of object **112**, computer **118** commands imaging device **114** via system bus **116** to capture raw image data representing object **112**. The captured raw image data is transferred over system bus **116** to computer **118** which performs various image processing functions on the image data before storing it in its internal memory. System bus **116** also passes various status and control signals between imaging device **114** and computer **118**.

Referring now to FIG. **2**, a block diagram of the preferred embodiment of imaging device **114** is shown. Imaging device **114** preferably comprises a lens **220** having an iris, a filter **222**, an image sensor **224**, a timing generator **226**, an analog signal processor (ASP) **228**, an analog-to-digital (A/D) converter **230**, an interface **232**, and one or more motors **234**.

U.S. patent application Ser. No. 08/355,031, entitled "A System and Method For Generating a Contrast Overlay as a Focus Assist for an Imaging Device," filed on Dec. 13, 1994 is incorporated herein by reference and provides a detailed discussion of the preferred elements of imaging device **114**. Briefly, imaging device **114** captures an image of object **112** via reflected light impacting image sensor **224** along optical path **236**. Image sensor **224** responsively generates a set of raw image data representing the captured image **112**. The raw image data is then routed through ASP **228**, AID converter **230** and interface **232**. Interface **232** has outputs for controlling ASP **228**, motors **234** and timing generator **226**. From interface **232**, the raw image data passes over system bus **116** to computer **118**.

Referring now to FIG. **3**, a block diagram of the preferred embodiment for computer **118** is shown. System bus **116** provides connection paths between imaging device **114**, power manager **342**, central processing unit (CPU) **344**, dynamic random-access memory (DRAM) **346**, input/output interface (I/O) **348**, read-only memory (ROM) **350**, and buffers/connector **352**. Removable memory **354** connects to system bus **116** via buffers/connector **352**. Alternately, camera **110** may be implemented without removable memory **354** or buffers/connector **352**.

Power manager **342** communicates via line **366** with power supply **356** and coordinates power management operations for camera **110** as discussed below in conjunction with FIGS. **7–10**. CPU **344** typically includes a conventional processor device for controlling the operation of camera **110**. In the preferred embodiment, CPU **344** is capable of concurrently running multiple software routines to control the various processes of camera **110** within a multi-threading environment. DRAM **346** is a contiguous block of dynamic memory which may be selectively allocated to various storage functions and is further discussed below in conjunction with FIGS. **4–11**.

I/O **348** is an interface device allowing communications to and from computer **118**. For example, I/O **348** permits an external host computer (not shown) to connect to and communicate with computer **118**. I/O **348** also permits a camera **110** user to communicate with camera **110** via a set of externally-mounted user controls and via an external LCD display panel. ROM **350** typically comprises a conventional nonvolatile read-only memory which stores a set of computer-readable program instructions to control the operation of camera **110**. ROM **350** is further discussed below in conjunction with FIG. **4**. Removable memory **354** serves as an additional image data storage area and is preferably a non-volatile device, readily removable and replaceable by a camera **110** user via buffers/connector **352**. Thus, a user who possesses several removable memories **354** may replace a full removable memory **354** with an empty removable memory **354** to effectively expand the picture-taking capacity of camera **110**. In the preferred embodiment of the present invention, removable memory **354** is typically implemented using a flash disk.

Power supply **356** supplies operating power to the various components of camera **110**. In the preferred embodiment, power supply **356** provides operating power to a main power bus **362** and also to a secondary power bus **364**. The main power bus **362** provides power to imaging device **114**, I/O **348**, ROM **350** and removable memory **354**. The secondary power bus **364** provides power to power manager **342**, CPU **344** and DRAM **346**.

Power supply **356** is connected to main batteries **358** and also to backup batteries **360**. In the preferred embodiment, a camera **110** user may also connect power supply **356** to an external power source. During normal operation of power supply **356**, the main batteries **358** provide operating power to power supply **356** which then provides the operating power to camera **110** via both main power bus **362** and secondary power bus **364**.

During a power failure mode in which the main batteries **358** have failed (when their output voltage has fallen below a minimum operational voltage level) the backup batteries **360** provide operating power to power supply **356** which then provides the operating power only to the secondary power bus **364** of camera **110**. Selected components of camera **110** (including DRAM **346**) are thus protected against a power failure in main batteries **358**.

Power supply **356** preferably also includes a flywheel capacitor connected to the power line coming from the main batteries **358**. If the main batteries **358** suddenly fail, the flywheel capacitor temporarily maintains the voltage from the main batteries **358** at a sufficient level, so that computer **118** can protect any image data currently being processed by camera **110** before shutdown occurs. The operation of power manager **342** and power supply **356** are further discussed below in conjunction with FIGS. **7–10**.

Referring now to FIG. **4**, a memory map showing the preferred embodiment of Read-Only-Memory (ROM) **350** is shown. In the preferred embodiment, ROM **350** includes control application **400**, toolbox **402**, drivers **404**, kernel **406** and system configuration **408**. Control application **400** comprises program instructions for controlling and coordinating the various functions of camera **110**. Toolbox **402** contains selected function modules including memory manager **410**, RAM spooler 1 (**412**), RAM spooler 2 (**414**), removable memory spooler 1 (**416**), removable memory spooler 2 (**418**), image processing and compression (IPC) **420** and file system **422**.

Memory manager **410** is controlled by control application **400** and responsively allocates DRAM **346** storage locations

depending upon the needs of computer **118** and the sets of received image data. File system **422** controls the file structure used in storing data onto DRAM **346**. In the preferred embodiment, DRAM **346** uses a standardized file system **422** which permits an external host computer system to readily recognize and directly access DRAM **346** via I/O **348**.

RAM spooler 1 (**412**), RAM spooler 2 (**414**), removable memory spooler 1 (**416**), removable memory spooler 2 (**418**), and image processing/compression (IPC) **420** are software routines used to spool, process, compress and store captured image data according to the present invention. These aforementioned software routines **412** through **420** are further discussed below in conjunction with FIG. **6**.

Drivers **404** control various hardware devices within camera **110** (for example, motors **234**). Kernel **406** provides basic underlying services for the camera **110** operating system. System configuration **408** performs initial start-up routines for camera **110**, including the boot routine and initial system diagnostics.

Referring now to FIG. **5**, a memory map showing the preferred embodiment of dynamic random-access-memory (DRAM) **346** is shown. In the preferred embodiment, DRAM **346** includes working memory **530**, RAM disk **532** and system area **534**. Working memory **530** includes frame buffers **536** (for initially storing sets of raw image data received from imaging device **114**) and image processing (IP) buffers **538** (for temporarily storing image data during the image processing and compression **420** process). Working memory **530** may also contain various stacks, data structures and variables used by CPU **344** while executing the software routines used within computer **118**.

RAM disk **532** is a memory area used for storing raw and compressed image data and typically is organized in a "sectored" format similar to that of conventional hard disk drives. In the preferred embodiment, RAM disk **532** uses a well-known and standardized file system to permit external host computer systems, via I/O **348**, to readily recognize and access the data stored on RAM disk **532**. System area **534** typically stores data regarding system errors (for example, why a system shutdown occurred) for use by CPU **344** upon a restart of computer **118**.

Referring now to FIG. **6**, a block diagram showing preferred data paths for transmitting image data between selected computer **118** components is shown. In FIG. **6**, frame buffer **536** receives and stores raw image data previously captured by imaging device **114**. Frame buffer **536** then transfers control of the raw image data to RAM spooler 1 (**412**) via line **610**. Alternately, if RAM disk **532** is full, frame buffer **536** may transfer control of the raw image data directly to image processing/compression (IPC) **420** using line **612**. If RAM spooler 1 (**412**) receives control of the raw image data, it then stores the raw image data into RAM disk **532** using line **614**.

Removable memory spooler 1 (**416**) may then access the raw image data from RAM disk **532** via line **616** and store it into removable memory **354** using line **618**. Alternately, if removable memory **354** is full or is not inserted, RAM disk **532** may provide the raw image data directly to IPC **420** using line **620**. If removable memory spooler 1 (**416**) stores the raw image data into removable memory **354**, then IPC **420** typically accesses the stored raw image data using line **622** and processes the raw data to obtain compressed image data.

IPC **420** may provide the compressed image data to RAM spooler 2 (**414**) using line **624**. If RAM disk **532** is temporarily full, IPC **420** may write the compressed data to temporary IP buffers **538** via line **628**. RAM spooler 2 (**414**) may then access the compressed image data via line **630** and write the accessed data into RAM disk **532** via line **614**. RAM spooler 2 (**414**) may also download the compressed image data to I/O interface **348** using line **632**. Once the compressed data is in RAM disk **532**, removable memory spooler 2 (**418**) then accesses the data via line **616** and writes the compressed data into removable memory **354** using line **618**. If removable memory **354** is not inserted, the compressed data remains on RAM disk **532**.

The present invention may thus process and store a sequence of captured images received from imaging device **114**. Although the above example traces the typical data path for a single captured image, the present invention may readily operate on a plurality of captured images progressing through various parts of computer **118** and therefore, multiple sets of image data may exist simultaneously within computer **118**.

Referring now to FIG. **7**, a flowchart of preferred method steps for performing a power-up sequence according to the present invention is shown. Initially, computer **118** waits **742** for a "wake up" signal which is typically generated in response to the activation of a camera **110** power on-off switch. After the "wake up" signal is generated, power manager **342** starts **744** power supply **356** in normal mode with the main batteries **358** providing operating power to power supply **356** which then responsively provides the operating power to main power bus **362** and also to secondary power bus **364**. If available operating power is insufficient, the startup process aborts.

CPU **344** then starts **746** normal DRAM **346** operation and determines **748** whether a RESUME bit has been set in power manager **342**. In the preferred embodiment, CPU **344** sets the RESUME bit in response to a power failure in order to indicate that CPU **344** should not be reset in a subsequent powerup of camera **110**. If the RESUME bit has been set, CPU **344** resumes **750** the current process which was interrupted by the intervening power failure.

If the RESUME bit has not been set, CPU **344** then determines **752** whether a MSAVE bit has been set in power manager **342**. In the preferred embodiment, CPU **344** sets the MSAVE bit to specify that RAM disk **532** contains image data that should be saved upon restart of computer **118**. If the MSAVE bit has not been set, computer **118** boots up **754** using the system configuration **408** routine and formats **756** a new RAM disk **532**. CPU **344** then runs **760** control application **400** for normal operation of camera **110**.

In step **752**, if the MSAVE bit has been set, then CPU **344** determines **764** whether the "instant on" feature is selected. The "instant on" feature provides a more rapid restart procedure, however, since the system is not initialized by a new bootup procedure, new system configuration files will not be accessed. If "instant on" is selected, CPU **344** restores **770** the externally-mounted LCD status display and runs **760** control application **400**. If "instant on" is not selected, computer **118** boots up **766** using the system configuration **408** routine and then recovers and mounts **768** RAM disk **532**. CPU **344** then runs **760** control application **400** for normal operation of camera **110**. The FIG. **7** process then goes **762** to the power status sequence of FIG. **8**.

Referring now to FIG. **8**, a flowchart of preferred method steps for performing a power status sequence according to the present invention is shown. Initially, CPU **344** determines **874** whether a user shutdown or a time out event has occurred. A user shutdown results from the deactivation of

the camera **110** power on-off switch by a user. A time out event powers down camera **110** in response to an extended period of non-use and is intended to prevent a user from accidentally leaving camera **110** powered up and thereby unnecessarily draining power from the main batteries **358**.

If CPU **344** detects **874** a user shutdown or a time out event, then the FIG. **8** process goes **876** to the normal power down sequence of FIG. **9**. If CPU **344** does not detect **874** a user shutdown or a time out event, then power manager **342** determines **878** whether the voltage of the main batteries **358** is greater than a selected threshold value. The threshold value may be selected as a minimum camera **110** operating voltage, or alternately may be incrementally higher that the minimum operating voltage. If the voltage of the main batteries **358** is not greater than the selected threshold value, then the FIG. **8** process goes **880** to the power failure sequence of FIG. **10**. However, if the voltage of the main batteries **358** is greater than the selected threshold value, then the FIG. **8** process returns to step **874** and CPU **344** checks again for a user shutdown or a time out event.

Referring now to FIG. **9**, a flowchart of preferred method steps for performing a normal power-down sequence according to the present invention is shown. Initially, CPU **344** waits **900** until processing of any image data is completed and then determines **902** whether any images are saved on RAM disk **532**. If no images are saved on RAM disk **532**, then CPU **344** clears **904** the MSAVE and RESUME bits in power manager **342**. CPU **344** then signals **906** power manager **342** to shut down and power manager **342** responsively directs power supply **356** to remove **908** power from the main power bus **362**. Next, CPU **344** halts **910** its operation and power manager **342** removes **912** power from CPU **344** and DRAM **346**. The FIG. **9** process then goes **914** to the power up sequence of FIG. **6**.

If CPU **344** determines, in step **902**, that images are saved on RAM disk **532**, then CPU **344** sets the MSAVE bit and clears **916** the RESUME bit in power manager **342**. Next, CPU **344** forces **918** a full refresh of DRAM **346** and then forces **920** DRAM **346** into a self-refresh mode. CPU **344** then signals **922** power manager **342** to shut down and power manager **342** responsively directs power supply **356** to remove **924** power from the main power bus **362**. Next, CPU **344** halts **926** its operation. The FIG. **9** process then goes **914** to the power up sequence of FIG. **6**.

Referring now to FIG. **10**, a flowchart of preferred method steps for performing a power failure sequence according to the present invention is shown. Initially, power manager **342** sets **1028** its PFAIL bit. The PFAIL bit records the occurrence of a power failure so that computer **118** software routines may subsequently access this information when needed. Next, power manager **342** responsively directs power supply **356** to remove **1030** power from the main power bus **362**. Power manager **342** then signals **1032** CPU **344** with an interrupt and CPU **344** responsively stops **1034** the current process.

Next, CPU **344** sets **1036** the RESUME bit in power manager **342**. CPU **344** then forces **1038** a full refresh of DRAM **346** and then forces **1040** DRAM **346** into a self-refresh mode. Next, CPU **344** signals **1042** power manager **342** to shut down and halts **1044** its operation. The camera **110** user may then replace **1046** the main batteries **358** and the FIG. **10** process goes **1048** to the power up sequence of FIG. **6**.

Referring now to FIG. **11**, a flowchart of preferred method steps for accessing DRAM **346** and/or removable disk **354** with an external host computer is shown. Initially, the external host computer requests **1150** access/ownership of one or more volumes stored on DRAM **346** and/or removable disk **354**. CPU **344** then determines **1152** whether DRAM **346** and/or removable disk **354** are currently being accessed. If DRAM **346** and/or removable disk **354** are currently being accessed, then CPU **344** refuses **1154** access/ownership of DRAM **346** and/or removable disk **354** to the external host computer.

However, if DRAM **346** and/or removable disk **354** are not currently being accessed, then CPU **344** grants **1156** access/ownership of DRAM **346** and/or removable disk **354** to the external host computer. The host computer then accesses **1158** DRAM **346** and/or removable disk **354**. Next, CPU **344** determines **1160** whether the host computer is still connected to DRAM **346** and/or removable disk **354**. If the host computer is still connected to DRAM **346** and/or removable disk **354**, then CPU **344** waits until the host computer is no longer connected to DRAM **346** and/or removable disk **354** and then reacquires **1162** access/ownership of DRAM **346** and/or removable disk **354** for camera **110**.

The invention has been explained above with reference to a preferred embodiment. Other embodiments will be apparent to those skilled in the art in light of this disclosure. For example, DRAM **346** may alternately store various types of information other than those described above in conjunction with the preferred embodiment. Further, the present invention may alternately provide backup power protection to various computer **118** components other than those described in the prior discussion of the preferred embodiment. Therefore, these and other variations upon the preferred embodiment are intended to be covered by the present invention, which is limited only by the appended claims.

What is claimed is:

1. A digital camera device comprising:

a CPU capable of concurrently processing multiple unprocessed images into processed images;

a memory device, coupled to said CPU, for storing sets of image data, comprising frame buffers for storing unprocessed image data and a random-access memory disk for storing unprocessed image data and processed image data;

a memory manager for allocating storage locations to store said sets of image data within said memory device;

a power management system, for monitoring a power supply to detect a power failure, configured to protect said sets of image data if said power failure is detected; and

an interface coupled to said memory device whereby an external host computer can access said sets of image data stored in said memory device.

2. The digital camera device of claim **1** further comprising a removable non-volatile memory coupled to said memory device for storing unprocessed image data and processed image data.

3. The digital camera device of claim **1** wherein said memory device uses a standardized file system for storing said sets of image data.

4. The digital camera device of claim **1** wherein said power management system comprises:

main batteries for providing operating power to said digital camera during normal operation;

backup batteries for providing operating power to said memory device during a power failure in said main batteries; and

a power manager device for sensing said power failure and responsively connecting said memory device to said backup batteries.

5. A method of implementing a digital camera device, comprising the steps of:

using a CPU to concurrently process multiple unprocessed images into processed images;

using a memory manager to allocate storage locations within a memory device comprising frame buffers for storing unprocessed image data and a random-access memory disk for storing unprocessed image data and processed image data;

storing sets of image data into said allocated storage locations;

using a power management system to monitor a power supply and detect a power failure;

using said power management system to protect said sets of image data from said power failure; and

communicating with said memory device via an interface whereby an external host computer can access said sets of image data.

6. The method of claim 5 further comprising a removable non-volatile memory coupled to said memory device for storing unprocessed image data and processed image data.

7. The method of claim 5 wherein said memory device uses a standardized file system for storing said sets of image data.

8. The method of claim 5 wherein said CPU can instruct said memory manager to allocate said storage locations.

9. A computer-readable medium comprising program instructions for causing a computer system to perform the steps of:

using a CPU to concurrently process multiple unprocessed images into processed images;

using a memory manager to allocate storage locations within a memory device comprising frame buffers for storing unprocessed image data and a random-access memory disk for storing unprocessed image data and processed image data;

storing sets of image data into said allocated storage locations;

using a power management system to monitor a power supply and detect a power failure;

using said power management system to protect said sets of image data from said power failure; and

communicating with said memory device via an interface whereby an external host computer can access said sets of image data.

10. The medium of claim 9 further comprising a removable non-volatile memory coupled to said memory device for storing unprocessed image data and processed image data.

11. The medium of claim 9 wherein said memory device uses a standardized file system for storing said sets of image data.

12. The medium of claim 9 wherein said CPU can instruct said memory manager to allocate said storage locations.

13. A digital camera device comprising:

CPU means for concurrently processing multiple unprocessed images into processed images;

memory manager means for allocating storage locations within a memory device comprising frame buffers for storing unprocessed image data and a random-access memory disk for storing unprocessed image data and processed image data;

means for storing sets of image data into said allocated storage locations; and

means for communicating with said memory device via an interface whereby an external host computer can access said sets of image data.

14. The digital camera device of claim 13 further comprising a removable non-volatile memory coupled to said memory device to store unprocessed image data and processed image data.

15. The digital camera device of claim 13 wherein said memory device uses a standardized file system for storing said sets of image data.

16. The digital camera device of claim 13 wherein said CPU can instruct said memory manager to allocate said storage locations.

* * * * *

Exhibit B

US00RE38911E

(19) **United States**

(12) **Reissued Patent**

Anderson et al.

(10) **Patent Number: US RE38,911 E**

(45) **Date of Reissued Patent: Dec. 6, 2005**

(54) **MODULAR DIGITAL IMAGE PROCESSING VIA AN IMAGE PROCESSING CHAIN WITH MODIFIABLE PARAMETER CONTROLS**

(75) Inventors: **Eric C. Anderson**, Gardnerville, NV (US); **Gary Chin**, Milpitas, CA (US)

(73) Assignee: **Apple Computer, Inc.**, Cupertino, CA (US)

(21) Appl. No.: **09/990,869**

(22) Filed: **Nov. 21, 2001**

Related U.S. Patent Documents

Reissue of:

(64) Patent No.: **5,991,465**
Issued: **Nov. 23, 1999**
Appl. No.: **08/705,325**
Filed: **Aug. 29, 1996**

(51) **Int. Cl.**[7] **G06K 9/54**; G05B 19/18; G06F 15/00

(52) **U.S. Cl.** **382/302**; 382/104; 382/303; 700/2; 700/4; 712/1; 712/32

(58) **Field of Search** 382/104, 167, 382/226, 227, 232, 234, 240, 248, 254, 256, 276, 302, 303, 304, 309, 310; 345/501–506, 645; 348/207.99, 207.1, 231.6, 251; 358/532; 700/2, 4; 712/1, 10, 11, 15–17, 32, 34, 35

(56) **References Cited**

U.S. PATENT DOCUMENTS

4,442,543	A *	4/1984	Sternberg et al.	382/303
4,794,524	A	12/1988	Carberry et al.	712/32
4,850,027	A	7/1989	Kimmel	382/303
4,860,375	A *	8/1989	McCubbrey et al.	382/303
4,949,390	A *	8/1990	Iverson et al.	382/303
5,237,626	A *	8/1993	Forslund et al.	382/256
5,283,671	A *	2/1994	Stewart et al.	358/532
5,315,700	A *	5/1994	Johnston et al.	345/502
5,365,596	A *	11/1994	Dante et al.	382/141
5,402,170	A	3/1995	Parulski et al.	348/211
5,452,375	A *	9/1995	Rousseau et al.	382/302
5,477,264	A	12/1995	Sarbadhikari et al.	348/231
5,513,382	A	4/1996	Williams	382/303
5,563,655	A	10/1996	Lathrop	348/231
5,568,192	A *	10/1996	Hannah	348/222
5,606,365	A	2/1997	Maurinus et al.	348/222
5,623,604	A *	4/1997	Russell et al.	395/712
5,631,974	A *	5/1997	Lau-Kee et al.	382/302
5,642,444	A *	6/1997	Mostafavi	382/303
5,661,823	A	8/1997	Yamauchi et al.	382/239
5,734,425	A	3/1998	Takizawa et al.	348/231
5,739,850	A	4/1998	Hori	348/231
5,764,817	A	6/1998	Suzuki et al.	382/302

* cited by examiner

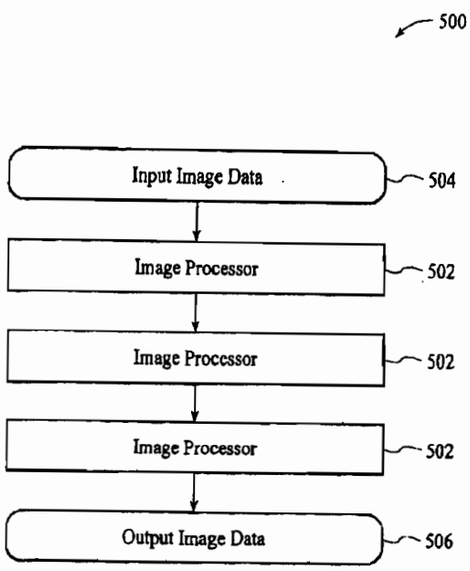
Primary Examiner—Daniel G. Mariam

(74) *Attorney, Agent, or Firm*—Fenwick & West LLP

(57) **ABSTRACT**

Aspects for allowing variably controlled alteration of image processing of digital image data in a digital image capture device include forming an image processing chain with two or more image processors to process digital image data, and providing one or more parametric controls within each of the two or more image processors. The aspects further include accessing chosen controls of the one or more parametric controls to modify the two or more image processors for alteration of the image processing.

41 Claims, 8 Drawing Sheets



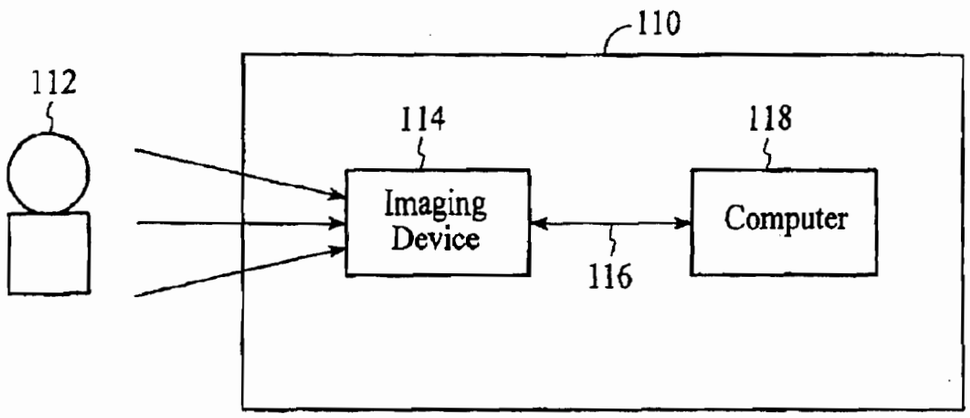


FIG. 1

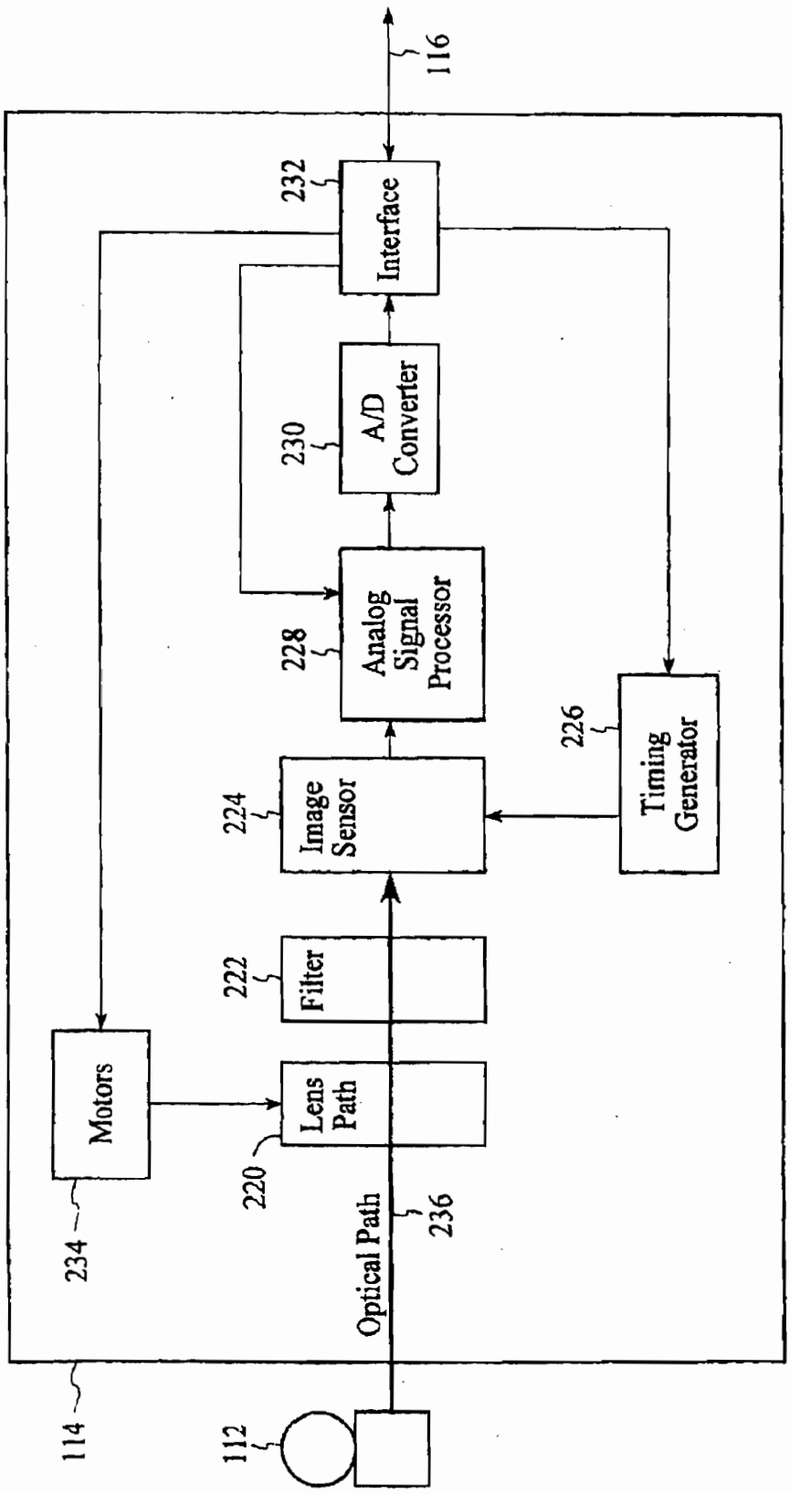

116
232
Interface
230
A/D Converter
228
Analog Signal Processor
226
Timing Generator
224
Image Sensor
222
Filter
Motors
Lens Path
220
234
Optical Path
236
114
112

FIG. 2

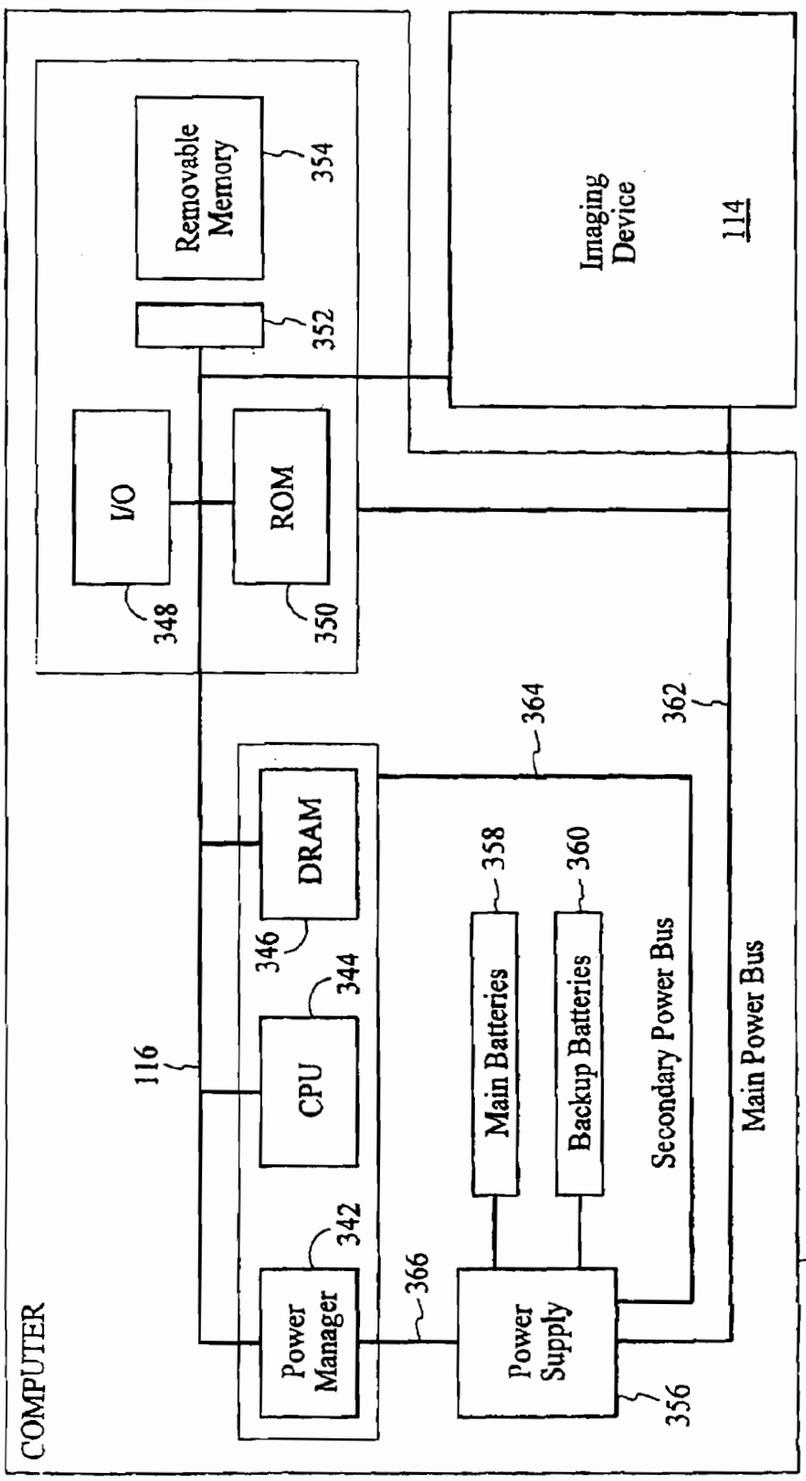

COMPUTER
116
Removable Memory
354
352
I/O
348
ROM
350
Imaging Device
114
DRAM
346
CPU
344
Power Manager
342
366
364
362
358
360
Main Batteries
Backup Batteries
Secondary Power Bus
Main Power Bus
Power Supply
356
118

FIG. 3

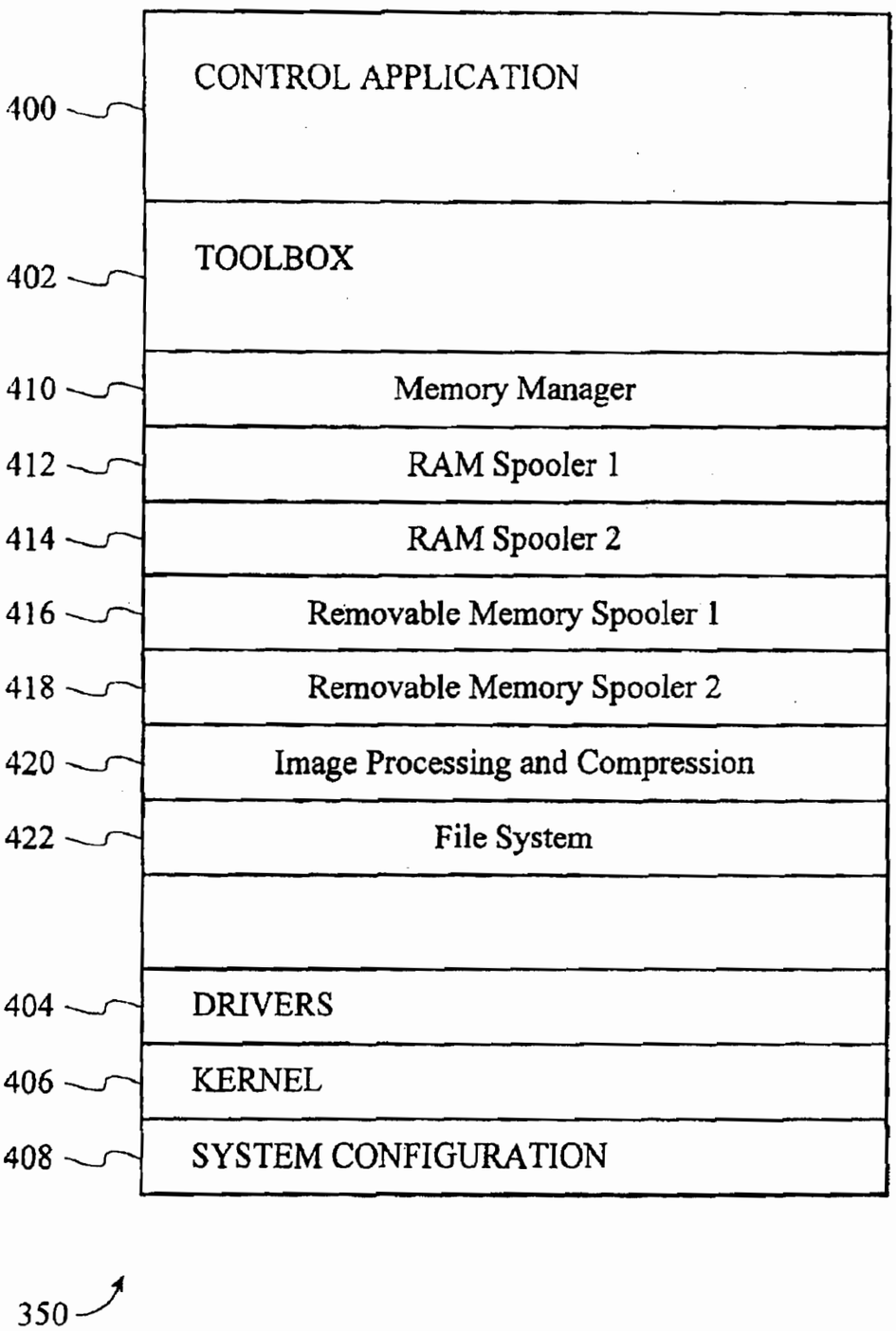

CONTROL APPLICATION
400
TOOLBOX
402
Memory Manager
410
RAM Spooler 1
412
RAM Spooler 2
414
Removable Memory Spooler 1
416
Removable Memory Spooler 2
418
Image Processing and Compression
420
File System
422
DRIVERS
404
KERNEL
406
SYSTEM CONFIGURATION
408
350

FIG. 4

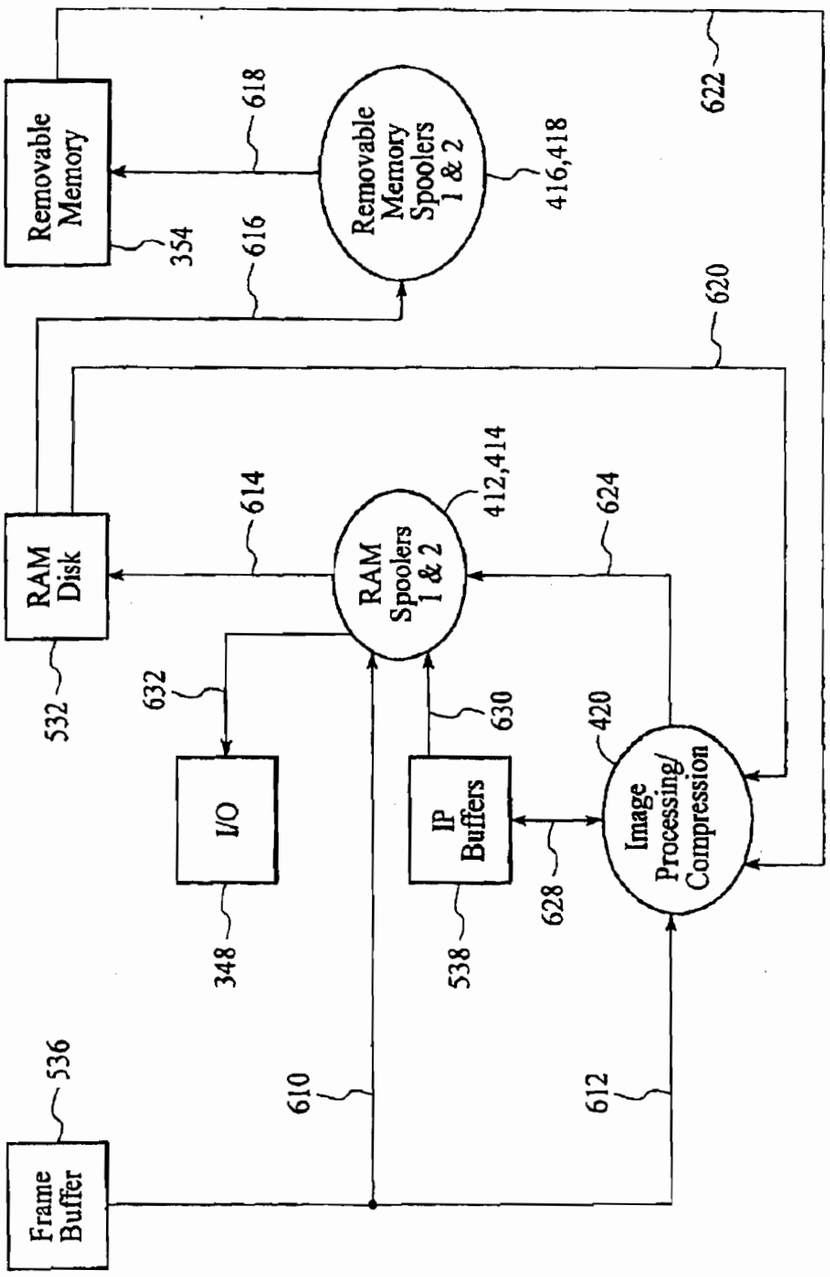

Removable Memory
354
618
616
Removable Memory Spoolers 1 & 2
416,418
622
620
RAM Disk
532
614
RAM Spoolers 1 & 2
412,414
624
632
630
I/O
348
IP Buffers
538
628
420
Image Processing/ Compression
Frame Buffer
536
610
612

FIG. 5

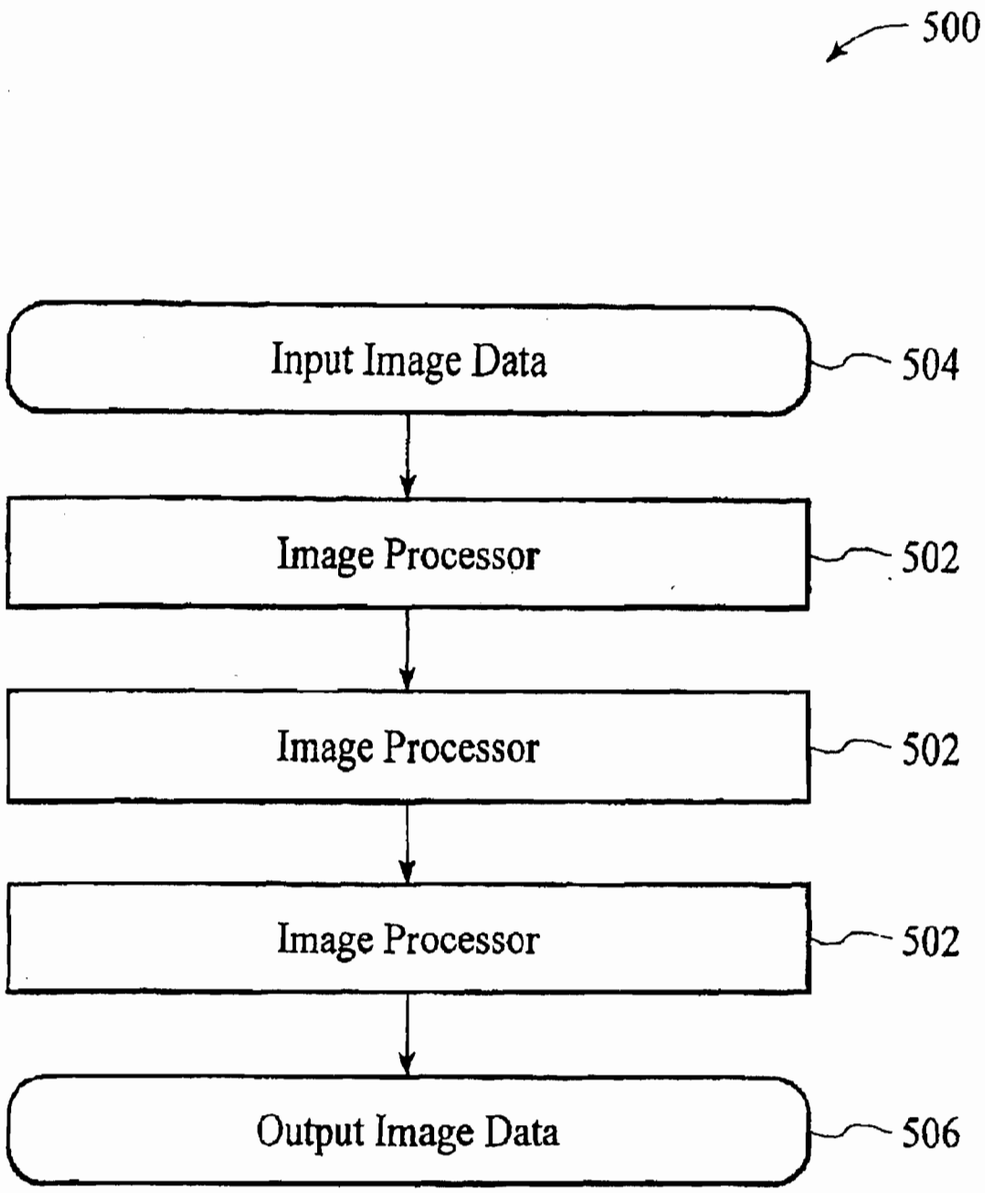

500
Input Image Data
504
Image Processor
502
Image Processor
502
Image Processor
502
Output Image Data
506

FIG. 6

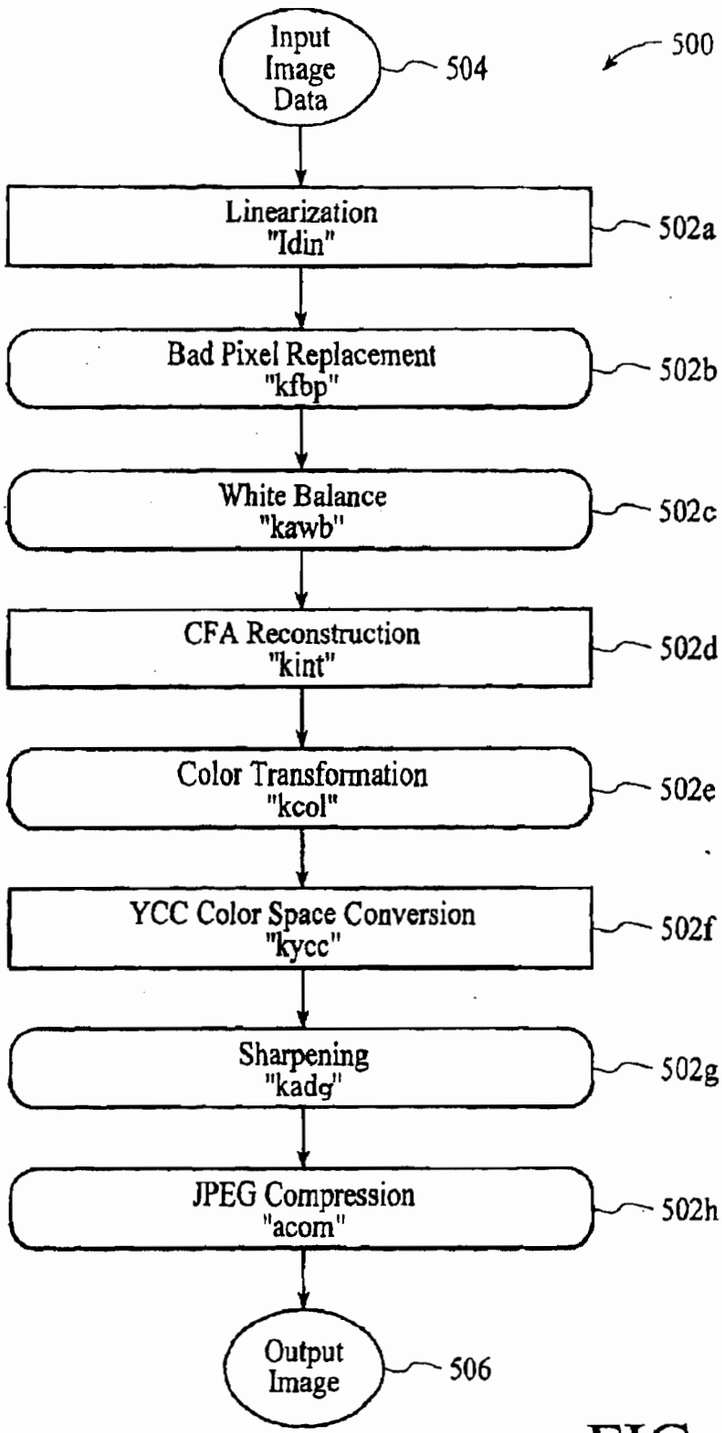

Input Image Data
504
500
Linearization "Idin"
502a
Bad Pixel Replacement "kfbp"
502b
White Balance "kawb"
502c
CFA Reconstruction "kint"
502d
Color Transformation "kcol"
502e
YCC Color Space Conversion "kycc"
502f
Sharpening "kadg"
502g
JPEG Compression "acom"
502h
Output Image
506

FIG. 7

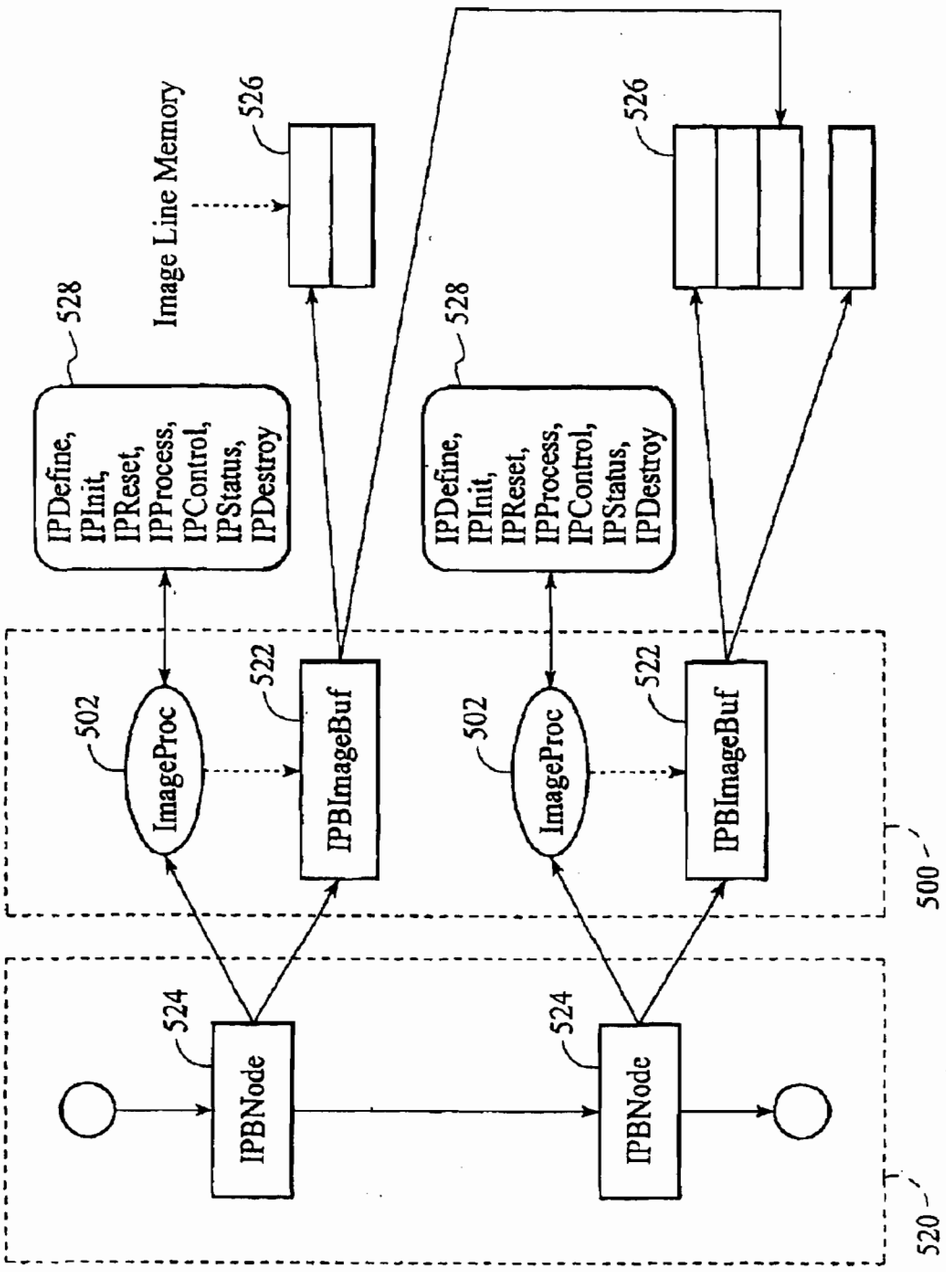


Image Line Memory
526
526
528
528
IPDefine,
IPInit,
IPReset,
IPProcess,
IPControl,
IPStatus,
IPDestroy
IPDefine,
IPInit,
IPReset,
IPProcess,
IPControl,
IPStatus,
IPDestroy
502
ImageProc
522
IPBImageBuf
502
ImageProc
522
IPBImageBuf
524
IPBNode
524
IPBNode
500
520

FIG. 8

MODULAR DIGITAL IMAGE PROCESSING VIA AN IMAGE PROCESSING CHAIN WITH MODIFIABLE PARAMETER CONTROLS

Matter enclosed in heavy brackets [ ] appears in the original patent but forms no part of this reissue specification; matter printed in italics indicates the additions made by reissue.

RELATED APPLICATIONS

The present invention is related to co-pending U.S. patent application, Ser. No. 08/705,619, filed on Aug. 29, 1996, entitled MODULAR DIGITAL IMAGE PROCESSING VIA AN IMAGE PROCESSING CHAIN, and assigned to the assignee of the present invention.

The present invention is also related to co-pending U.S. patent application, Ser. No. 08/705,588 filed on [Sep.] *Aug.* 29, 1996 (*which is now U.S. Pat. No. 6,157,394 issued on Dec. 5, 2000*), entitled FLEXIBLE DIGITAL IMAGE PROCESSING VIA AN IMAGE PROCESSING CHAIN WITH MODULAR IMAGE PROCESSORS, and assigned to the assignee of the present invention.

FIELD OF THE INVENTION

The present invention relates to digital image data processing, and more particularly to modular digital image data processing with modifiable parameter control.

BACKGROUND OF THE INVENTION

Modern digital cameras typically include an imaging device which is controlled by a computer system. The computer system accesses raw image data captured by the imaging device and then processes and compresses the data before storing the compressed data into an internal memory. The conventional digital camera captures image data and then remains unusable until the data is completely processed and stored into internal flash memory.

In processing image data, typical digital cameras operate with exclusive and specific image processing. Thus, all the potential manipulation on image data, such as linearization, sharpening, and compression, occur as a result of isolated preset programming and/or specifically designed hardware.

While some level of manipulation of image data is achieved with the programming or hardware, attempts to alter and improve the processing are hampered by the rigid structure of using a single file/specific components. Furthermore, camera functionality remains tied to technology available at the time of the design and is not readily replaced and updated as technology improves. Accordingly, a need exists for a more flexible, modular approach for processing digital image data that provides enhanced digital image output through an adaptable image processing system.

SUMMARY OF THE INVENTION

Accordingly, the present invention meets these needs and provides a method and system for allowing variably controlled alteration of image processing of digital image data in a digital image capture device. In a method aspect, the method includes forming an image processing chain with two or more image processors to process digital image data, and providing one or more parametric controls within each of the two or more image processors. The method further includes accessing chosen controls of the one or more parametric controls to modify the two or more image processors for alteration of the image processing.

In a system aspect, the system includes a digital image capture device, the digital image capture device capable of processing digital image data through two or more image processors, the two or more image processors have one or more parametric controls, and a central processing unit. The central processing is included within the digital image capture device and capable of linking the two or more image processors to form an image processing chain. The central processing unit further facilitates access of chosen controls of the one or more parametric controls for modification of the two or more image processors and alteration of the image processing.

With the present invention, processing of digital image data occurs with a linked series of image processors. Each of the image processors performs some level of manipulation of the digital image data. The separation of digital image processing into a series of image processors allows a more modular approach to processing digital image data. Further, the present invention uniquely allows modification of the series through deletion of an image processor, insertion of a different image processor, or replacement of an existing image processor. In addition, aspects of an image processor, including parameter control values, are alterable in accordance with a preferred embodiment to allow greater adaptability to user-specific design preferences. Enhancements and changes to the chain are therein easily achieved, allowing greater flexibility and more convenient upgrading of digital image processing.

These and other advantages of the aspects of the present invention will be more fully understood in conjunction with the following detailed description and accompanying drawings.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. **1** is a block diagram of a digital camera that operates in accordance with the present invention.

FIG. **2** is a block diagram of the preferred embodiment for the imaging device of FIG. **1**.

FIG. **3** is a block diagram of the preferred embodiment for the computer of FIG. **1**.

FIG. **4** is a memory map showing the preferred embodiment of the read only memory (ROM) of FIG. **3**.

FIG. **5** is a block diagram showing preferred data paths for transmitting image data between components of the FIG. **3** computer.

FIG. **6** illustrates an image processing chain of three image processors.

FIG. **7** illustrates a more specific example of the image processing chain.

FIG. **8** illustrates an image processing backplate in conjunction with the image processing chain of FIG. **6**.

DETAILED DESCRIPTION

The present invention relates to a flexible, modular approach to processing of digital image data. The following description is presented to enable one of ordinary skill in the art to make and use the invention and is provided in the context of a patent application and its requirements. Various modifications to the preferred embodiment and the generic principles and features described herein will be readily apparent to those skilled in the art.

Although the following describes processing of digital image data captured through a digital camera device, it is meant as an illustrative embodiment of the features of the

present invention. The present invention is equally capable of utilization with other devices that perform digital image data capture and processing, including, but not limited to, computer systems, including those used to capture digital images accessible from Internet sites and image scanner equipment. Further, the data structures and commands discussed with reference to a preferred embodiment are suitably included as part of high level code used directly by one or more applications that is readily achieved through the use of C, C++, or other similar programming language, and stored on a computer readable medium.

A digital camera architecture has been disclosed in co-pending U.S. patent application Ser. No. 08/666,241, entitled "A System And Method For Using A Unified Memory Architecture To Implement A Digital Camera Device," filed on Jun. 20, 1996, and assigned to the Assignee of the present application. The Applicant hereby incorporates the co-pending application by reference, and reproduces portions of that application herein with reference to FIGS. 1–5 for convenience.

Referring now to FIG. 1, a block diagram of a camera **110** is shown according to the present invention. Camera **110** preferably comprises an imaging device **114**, a system bus **116** and a computer **118**. Imaging device **144** is optically coupled to an object **112** and electrically coupled via system bus **116** to computer **118**. Once a photographer has focused imaging devices **114** on object **112** and, using a capture button on some other means, instructed camera **110** to capture an image of object **112**, computer **118** commands imaging device **114** via system bus **116** to capture raw image data representing object **112**. The captured raw image data is transferred over system bus **116** to computer **118** which performs various image processing functions on the image data before storing it in its internal memory. System bus **116** also passes various status and control signals between imaging device **114** and computer **118**.

Referring now to FIG. 2, a block diagram of the preferred embodiment of imaging device **114** is shown. Imaging device **114** preferably comprises a lens **220** having an iris, a filter **222**, an image sensor **224**, a timing generator **226**, an analog signal processor (ASP) **228**, an analog-to-digital (A/D) converter **230**, an interface **232**, and one or more motors **234**.

U.S. patent application Ser. No. 08/355,031, entitled "A System and Method For Generating a Contrast Overlay as a Focus Assist for an Imaging Device," filed on Dec. 13, 1994, is incorporated herein by reference and provides a detailed discussion of the preferred elements of imaging device **114**. Briefly, imaging device **114** captures an image of object **112** via reflected light impacting image sensor **224** along optical path **236**. Image sensor **224** responsively generates a set of raw image data representing the captured image **112**. The raw image data is then routed through ASP **228**, A/D converter **230** and interface **232**. Interface **232** has outputs for controlling ASP **228**, motors **234** and timing generator **226**. From interface **232**, the raw image data passes over system bus **116** to computer **118**.

Referring now to FIG. 3, a block diagram of the preferred embodiment for computer **118** is shown. System bus **116** provides connection paths between imaging device **114**, power manager **342**, central processing unit (CPU) **344**, dynamic random-access memory (DRAM) **346**, input/output interface (I/O) **348**, read-only memory (ROM) **350**, and buffers/connector **352**. Removable memory **354** connects to system bus **116** via buffers/connector **352**. Alternately, camera **110** may be implemented without removable memory **354** or buffers/connector **352**.

Power manager **342** communicates via line **366** with power supply **356** and coordinates power management operations for camera **110**. CPU **344** typically includes a conventional processor device for controlling the operation of camera **110**. In the preferred embodiment, CPU **344** is capable of concurrently running multiple software routines to control the various processes of camera **110** within a multi-threading environment. DRAM **346** is a contiguous block of dynamic memory which may be selectively allocated to various storage functions.

I/O **348** is an interface device allowing communications to and from computer **118**. For example, I/O **348** permits an external host computer (not shown) to connect to and communicate with computer **118**. I/O **348** also permits a camera **110** user to communicate with camera **110** via an external user interface and via an external display panel, referred to as a view finder.

ROM **350** typically comprises a conventional nonvolatile read-only memory which stores a set of computer-readable program instructions to control the operation of camera **110**. ROM **350** is further discussed below in conjunction with FIG. 4. Removable memory **354** serves as an additional image data storage area and is preferably a non-volatile device, readily removable and replaceable by a camera **110** user via buffers/connector **352**. Thus, a user who possesses several removable memories **354** may replace a full removable memory **354** with an empty removable memory **354** to effectively expand the picture-taking capacity of camera **110**. In the preferred embodiment of the present invention, removable memory **354** is typically implemented using a flash disk.

Power supply **356** supplies operating power to the various components of camera **110**. In the preferred embodiment, power supply **356** provides operating power to a main power bus **362** and also to a secondary power bus **364**. The main power bus **362** provides power to imaging devices **114**, I/O **348**, ROM **350** and removable memory **354**. The secondary power bus **364** provides power to power manager **342**, CPU **344** and DRAM **346**.

Power supply **356** is connected to main batteries **358** and also to backup batteries **360**. In the preferred embodiment, a camera **110** user may also connect power supply **356** to an external power source. During normal operation of power supply **356**, the main batteries **358** provide operating power to power supply **356** which then provides the operating power to camera **110** via both main power bus **362** and secondary power bus **364**.

During a power failure mode in which the main batteries **358** have failed (when their output voltage has fallen below a minimum operational voltage level) the backup batteries **360** provide operating power to power supply **356** which then provides the operating power only to the secondary power bus **364** of camera **110**. Selected components of camera **110** (including DRAM **346**) are thus protected against a power failure in main batteries **358**.

Power supply **356** preferably also includes a flywheel capacitor connected to the power line coming from the main batteries **358**. If the main batteries **358** suddenly fail, the flywheel capacitor temporarily maintains the voltage from the main batteries **358** at a sufficient level, so that computer **118** can protect any image data currently being processed by camera **110** before shutdown occurs.

Referring now to FIG. 4, a memory map showing the preferred embodiment of ROM **350** is shown. In the preferred embodiment, ROM **350** includes control application **400**, toolbox **402**, drivers **404**, kernal **406** and system

configuration **408**. Control application **400** comprises program instructions for controlling and coordinating the various functions of camera **110**. Toolbox **402** contains selected function modules including memory manager **410**, RAM spooler 1 **(412)**, RAM spooler 2 **(414)**, removable memory spooler 1 **(416)**, removable memory spooler 2 **(418)**, image processing and compression **420** and file system **422**.

Referring now to FIG. **5**, a block diagram showing preferred data paths for transmitting image data between selected computer **118** components is shown. In FIG. **5**, frame buffer **536** receives and stores raw image data previously captured by image device **114**. Frame buffer **536** then transfers control of the raw image data to RAM spooler 1 **(412)** via line **610**. Alternatively, if RAM disk **532** is full, frame buffer **536** may transfer control of the raw image data directly to image processing/compression **420** using line **612**. If RAM spooler 1 **(412)** receives control of the raw image data, it then stores the raw image data into RAM disk **532** using line **614**.

Removable memory spooler 1 **(416)** may then access the raw image data from RAM disk **532** via line **616** and store it into removable memory **354** using line **618**. Alternatively, if removable memory **354** is full or is not inserted, RAM disk **532** may provide the raw image data directly to image processing/compression **420** using line **620**. If removable memory spooler 1 **(416)** stores the raw image data into removable memory **354**, then image processing/compression **420** typically accesses the stored raw image data using line **622**.

LINKING IMAGE PROCESSORS FOR FORMING IMAGES

In the preferred embodiment, image processing and compression **420** occurs via an image processing chain (IPC). For purposes of this discussion, the IPC preferably refers to a software process that manipulates image data in a stage by stage fashion. As shown in FIG. **6**, an IPC **500** is suitably composed of a sequence of image processors **502** with each image processor **502** performing a particular type of image transformation. The input image data **504** is suitably received from a single image source and output as output image data **506** into a single image destination. Image processors suitably refer to software modules that apply algorithms on image data to obtain a special image processing result, specific examples of which are described below with reference to FIG. **7**.

FIG. **7** illustrates the IPC **500** with several examples of the image processor **502** capable for utilization as the IPC **500**. For each of the image processors **502**, an unambiguous image data format is specified for the input and output data. When the input and output image data formats are the same, the image processor **502** is considered non-transforming, examples of which are represented by the rounded boxes in FIG. **7**. Conversely, image processors **502** that do not have the same input and output data formats are suitably considered transforming, e.g., the rectangles of FIG. **7**. Although the following description of FIG. **7** is given with a particular order and series of image processors for image processing to occur in a sequential and serial manner, it should be appreciated that in the preferred embodiment, any number of non-transforming image processors may be chained between two separate transforming image processors. Further, brief descriptions of the type of image processing capable by each image processor **502** are included as examples. However, the details of such processing are not included in the present discussion and are considered to be well understood by those skilled in the art. Thus, image processing through the use of other image processors in the IPC **500** is within the spirit and scope of the present invention.

The image processors **502** suitably include a first image processor **502**a for linearization of the input image data **504**. By way of example, linearization refers to a straightforward conversion of the image data from an eight-bit non-linear space to sixteen-bit linear space. As a more specific example, input pixels stored as eight bit compressed Bayer pattern image data are converted through linearization image processor **502**a into sixteen bit extended Bayer pattern image data.

A next suitable image processor **502**b is a bad pixel replacement processor. Bad pixel replacement suitably occurs through interpolation of the neighborhood pixels around the defective CCD pixels. The processing by image processor **502**b capably receives and outputs pixel data in sixteen-bit linear space Bayer format.

As a next image processor **502**, white balance processor **502**c performs white balance image processing. Pixel data received and output by the white balance image processor **502**c are approximately stored in sixteen-bit linear space Bayer format.

A fourth image processor **502**d preferably performs image color or color filter array data (CFA) reconstruction. By way of example, the CFA reconstruction image processor **502**d suitably achieves an interpolation operation to convert sixteen-bit Bayer CFA pattern CCD data into a forty-eight bit extended RGB image.

Following CFA reconstruction image processor **502**d, color transformation image processor **502**e is included. An appropriate color transformation image processor **502**e employs a color correction matrix, such as to convert from device-dependent camera color space to device-independent linear CCIR709 color space. Preferably, the input and output pixel data is stored in forty-eight bit extended RGB format.

As a next image processor, YCC color space transformation image processor **502**f is included. The YCC color space conversion image processor **502**f suitably uses CCIR 601-2 specification to create an eight-bit YCrCb image from an RGB image. Input pixel data to image processor **502**f is suitably given in forty-eight bit extended RGB format with output pixel data in twenty-four bit YCrCb444 format.

Two additional image processors **502** in the IPC **500** include sharpening image processor **502**g and JPEG compression image processor **502**h. Sharpening image processor **502**g suitably receives input pixel data in twenty-four bit YCrCb format and outputs pixel data in the same format after performing sharpening operations. Parameter control of the sharpening suitably occurs with a range of values for the sharpening operation.

The JPEG compression image processor **502**h suitably performs JFIF base line image compression. Input pixel data in twenty-four bit YCrCb444 format is output from image processor **502**h as compressed and subsampled YCC format, 48-bit YCrCb411 per 4-pixel data. Two forms of parameter controls are achieved via image processor **502**h to both control the degree of compression, e.g., maximum to normal to lossless, and to identify data as color or grayscale.

Coordination of the image processors **502** to form the IPC **500** is preferably done via an image processing backplane (IPB). In a preferred embodiment, the image processing backplane provides processing support in a broad manner to allow varying algorithms to be incorporated as image processors **502**. The features of the processing support by the IPB are described in more detail with reference to FIG. **8** and include performing image scan line buffer input/output (I/O), IPC construction and connection, image processor parameter control setting, single pass through image data, procedural interface to the image processors, circular data

pipeline support, and ring-pixel handling, with minimal memory requirements and overheads.

FIG. **8** illustrates schematically an IPB **520** in conjunction with an IPC **500** including two image processors **502**. Suitably, internal data structures, e.g., IPBImageBuf**522** and IPBNode **524**, are maintained by the IPB **520** for storing information related to the image processors **502** during processing and are connected indirectly in terms of data flow, as indicated by the dashed arrows in FIG. **8**. The information maintained by the data structures **522** and **524** preferably includes locations of the input and output line buffers, and internal state and functional routine entry pointers of each image processor **502**. Thus, data structure **522** capably contains pointers pointing to image scan line buffers **526** that are used to store input and output image data in formats suitably determined during installation of an image processor **502**. More particularly, the image scan line buffers **526** preferably store one or more image scan lines, i.e., the lines of data forming a data pipeline that consists of the minimum number of lines required by an image processor **502**.

For image processors **502**, processing suitably occurs with a data pipeline that contains a single image scan line, i.e., an image pixel line in the fast scan direction from left to right. However, some image processors **502**, such as compression image processors (e.g., **502**h, FIG. **7**), utilize more than one scan line during processing to take neighboring effects into account. When more than one scan line is needed by an image processor **502**, a data pipeline is suitably defined for convenience at the input end of the image processor **502**. For purposes of this discussion, a data pipeline refers to a minimum collection of image scan lines required by an image processor **502**. Generally, a data pipeline includes an image scan line currently being processed, and some number of image scan lines prior to ('lookback') and/or after ('lookahead') the current image scan line. Suitably, access to the data pipeline occurs via a circular array of buffer pointers, so that after each processing iteration of the image processor **502**, the pointers in the array are circularly rotated, as is well understood by those skilled in the art. In contrast to prior devices that typically require large amounts of memory to perform image data manipulations, the image scan line buffers provide sufficient memory to perform processing one scan line at a time, thus reducing the overall memory requirements without reducing processing capabilities.

Preferably, the data pipeline required by an image processor **502** is indicated during the installation of the image processor **502** in the IPC **500**. Installation of an image processor **502** suitably occurs when the camera first starts up with an IPC **500** constructed from all of the default image processors **502** stored in the system ROM. Suitable functions to coordinate the construction and deconstruction of the IPC **500** include four functions, an initialization function, e.g., IPCInit, an installation function, e.g., IPCInstallImageProcessor, a connection function, e.g., IPCConnect, and a destruction function, e.g., IPCDestroy.

The IPC initialization function is called to create a new IPC **500**. A suitable default IPC **500** converts raw CCD capture data into a JPEG compressed image. Preferably, the IPC initialization function returns a reference to a new image processing chain, identifies types of image processors included in the chain, and specifies a maximum expected width in pixels to be sent through the IPC, where the maximum width includes ring-pixels, which refer to supplementary image data at each side of the image required by an image processor to perform a particular algorithm.

The IPC installation function is called by an image processing application to the IPB **520** to install the image processor **502** into the IPC **500**. Preferably, the installation function specifies an IPC reference number, as identified in the initialization function, and provides pointers to the seven functional routine entries, as discussed hereinbelow, of the image processor being installed.

The IPC connection function specifies an IPC reference number, and signals to the IPB **520** that all image processors **502** have been installed and that the IPC **500** contains all the required image processors **502** to perform image processing. The IPC destruction function specifies an IPC reference number and is called to destroy an IPC **500**. Although a default camera IPC **500** is unlikely to be destroyed, other IPCs added to a camera for other purposes by functions in accordance with a preferred embodiment and discussed in more detail hereinbelow, are suitably destroyed with this function.

MODULARITY OF IMAGE PROCESSORS THROUGH FLEXIBLE UPDATING OF AN IPC

Alterations to an existing IPC **500** readily occur in a preferred embodiment through an update function, e.g., IPBUpdateDefaultPC, that specifies the IPC reference number for the IPC **500** being updated/modified. Updating of an IPC **500** includes insertion of an image processor **502** to the IPC **500**, deletion of an image processor **502** from the IPC **500**, or replacement of an image processor **502** with an alternate image processor **502**. Preferably, the default IPC **500** is updated via an image processor module on a storage device, e.g., removable memory, RAM disk, or internal memory. The image processor module suitably contains one or more plug-in image processors that each have one additional function, e.g., IPMPlugInProc, that defines the updating strategy, the signature of the target image processor to be updated, and pointers of the seven basic functions of an image processor, as described hereinbelow. Lack of identification of valid target image processors in an IPC or lack of match between the format of the output of one image processor and input of a next image processor chained together preferably results in cancellation of the updating attempt and restoration of the default IPC **500**.

Defining an image processor **502**, for use in a default IPC **500** or as an updating image processor, suitably occurs through seven functional routines or procedures, as indicated by block **528** in FIG. **8**. A definition function, e.g., IPDefineProc, allows an image processor **502** to specify its characteristics. It is appropriately called by the IPB **520** when the image processor **502** is installed into the IPC **500** to identify the characteristics of the image processor **502**. By way of example, for an image processor **502** that performs color correction via a 3×3 matrix, input and output formats of 48 bit extended RGB are capably identified by the definition function. Further characteristics identified include the configuration of the data pipeline associated with the image processor **502**, the number of ring-pixels, and the number of parameter controls.

An initialization, e.g., IPInitProc, appropriately allows an image processor **502** to allocate any internal storage it might need when processing an image. It is suitably called by the IPB **520** only once when the image processor **502** is installed into the IPC **500**. Subsequent calls to the other five functions described below then pass the internal storage space allocated by the IPInitProc as an argument. Further identified by the initialization function is the maximum width specification of an image scan line in pixels that is expected at the input, including ring-pixels at both the left and right sides. In the example of defining the color correction image

processor, the initialization function capably identifies a memory location storing a pointer to needed 3×3 matrix constant value internal variables, and the maximum width.

Two functions, a control function, e.g., IPControlProc, and a status function, e.g., IPStatusProc, deal with parameter controls of an image processor **502**. Preferably, parameter controls for an image processor **502** each have a unique 4-character tag that is registered to avoid conflict. Parameter control values include two types, a range type and an enumerated list type. Range types of parameter control values are appropriately confined between the minimum and maximum settings for the range. Enumerated list parameter control values assign different enumerated numbers to different settings with a 32-character null terminated string used to provide an ASCII name for each enumerated list number. Examples of parameter controls include sharpening values (range type), color specification control values (range type), and compression control values (enumerated list type).

The control function, IP ControlProc, is called by the IPB **520** to control the processing parameters one parameter control at a time and only before a reset function, e.g., IPReset, call for every image to be processed. The status function, IPStatusProc, allows an image processor **502** called by the IPB **520** to determine any parameter kind, values types, factory default parameter setting, and current parameter setting of an image processor **502**. No-operation routines are provided when the image processor does not support any parameter settings, such as in the example of the color correction image processor.

The reset function, IPResetProc, suitably allows IPB **520** to signal an image processor **502** to reset any internal variables used by the image processor **502** before every image is processed. With the color correction image processor example, no operation routines are provided, since no local variables need to be reset. A process function, e.g., IPProcessProc, suitably allows an image processor **502** to process image data one scan line at a time. It is suitably called whenever a data pipeline for the image processor **502** fills up. Thus, the operations for performing the 3×3 matrix manipulation in the color correction image processor example, are specified with the process function. A destruction function, e.g., IPDestroyProc, suitably allows an image processor to deallocate any internal storage allocated at initialization. It is appropriately called when the IPC **500** containing the image processor **502** is being removed. In response to this call, the image processor **502** preferably deallocates internal storage allocated in the initialization function call.

Entry points to these seven functional routines for an image processor **502**, stored in an external data structure, e.g., Functions, as well as the characteristics of the image processor **502**, are suitably stored in an internal data structure, e.g., ImageProc, by the IPB **520**. Once the image processor **502** are defined through the seven functional routines and connected in an IPC **500**, the IPB **520** suitably facilitates image processing operations by managing image buffer I/O, and activation of each image processor **502** as soon as enough input data has been collected. The information for the image data processed is suitably stored in a data structure, e.g., ImageInfo, including raw image size captured by a camera CCD, final processed output image size, bad pixel locations, etc.

MODIFIABLE PARAMETER CONTROL OR IMAGE PROCESSOR IN AN IPC

In a preferred embodiment, the IPB **520** further provides routines to allow exchanges of parameter control settings by an external mechanism, such as a control application **400** (FIG. **4**). These functions include parameter control capability and value determination functions, e.g., IPBGetParameterCapability, IPBGetDefaultParameter, and IPBGetParameter. Also included are functions for setting or restoring parameter control values, e.g., IPBSetDefaultParameter, IPBSetParameter, and IPBRestoreParameter. Preferably, for the parameter control value determination functions, an IPC **500** and the number of parameters requested are identified, as well as identification of a pointer to an array of parameter tags, a pointer to a memory location used to store a pointer for the parameter settings returned, and a pointer to a memory location where the number of bytes of parameter control values are stored. Similarly, for the capability determination function, an IPC is specified, the number of parameters requested is specified, a pointer to an array of parameter tags is specified, a pointer to a memory location used to store a pointer for the parameter capability information returned is specified, and a pointer to a memory location where the number of bytes of parameter capability information is stored is specified. Thus, access to the parameter controls managed by an IPC **500** are available, as well as current values, device dependent factory default values, and user-specified default values.

For the parameter control value setting functions, preferably identified by the functions are an IPC, a number of parameter control values to be set, and a pointer to a list of parameter tags and either a current value or a user default value that are to be set by the function. In addition, the set parameter function appropriately allows all parameters not listed in the specified parameter value list to be reset to their user default value when a Boolean variable is set. The restoration function similarly identifies an IPC and a number of parameters to be requested, provides a pointer to an array of parameter tags, and selects the type of parameter defaults, i.e., user-specified or device dependent factory, being reset through a Boolean variable. These functions therefore provide convenient accessibility to allow alteration of parameter control values in an IPC **500**. Greater flexibility for adjusting an image processor **502** within an IPC **500** is advantageously provided.

Although the present invention has been described in accordance with the embodiments shown, one of ordinary skill in the art will recognize that there could be variations to the embodiment and those variations would be within the spirit and scope of the present invention. Accordingly, many modifications may be made by one of ordinary skill without departing from the spirit and scope of the present invention, the scope of which is defined by the following claims.

What is claimed is:

1. A method for allowing variably controlled alteration of image processing of digital image data in a digital image capture device, the method comprising:

forming an image processing chain with two or more image processors, the two or more image processors being stored in memory, wherein said processors are software modules and each performing a particular type of image transformation, to process digital image data;

providing one or more parametric controls that are uniquely identified and within each of the two or more image processors; and

accessing chosen controls of the one or more parametric controls within each of the two or more image processors to modify the two or more image processors for alteration of the image processing.

2. The method of claim **1** wherein the step of accessing further comprises altering a default value of the one or more parametric controls.

3. The method of claim 2 wherein the step of altering further comprises setting the default value to a desired value.

4. The method of claim 2 wherein the step of altering further comprises resetting the default value to a device dependent factory value.

5. The method of claim 1 wherein the step of accessing further comprises determining current values of the one or more parametric controls.

6. The method of claim 1 wherein the step of accessing further comprises determining default values of the one or more parametric controls.

7. The method of claim 1 wherein the step of accessing further comprises determining parametric control capabilities of the one or more parametric controls.

8. The method of claim 7 wherein the step of determining parametric control capabilities further comprises providing values, value types, and device dependent factory default values.

9. The method of claim 1 wherein the two or more image processors further comprise a sharpening image processor and a compression image processor.

10. The method of claim 9 wherein the sharpening image processor provides a sharpening parametric control.

11. The method of claim 10 wherein the sharpening parametric control comprises a range type of control.

12. The method of claim 9 wherein the compression image processor provides a compression parametric control and a color specification control.

13. The method of claim 12 wherein the compression parametric control comprises an enumerated list type of control.

14. The method of claim 12 wherein the color specification parametric control comprises a range type of control.

15. A system for allowing variably controlled alteration of image processing of digital image data, the system comprising:

a digital image capture device, the digital image capture device capable of processing digital image data through two or more image processors, the two or more image processors being stored in memory, wherein said processors are software modules and each performing a particular type of image transformation and having one or more parametric controls that are uniquely identified; and

a central processing unit within the digital image capture device and capable of linking the two or more image processors to form an image processing chain, wherein the central processing unit facilitates access of chosen controls of the one or more parametric controls within each of the two or more image processors for modification of the two or more image processors and alteration of the image processing.

16. The system of claim 15 wherein the two or more image processors further comprise a sharpening image processor and a compression image processor.

17. The system of claim 15 wherein the central processing unit facilitates altering a default value of the one or more parametric controls.

18. The system of claim 15 wherein the central processing unit facilitates setting the default value to a desired value.

19. The system of claim 15 wherein the central processing unit facilitates resetting the default value to a device dependent factory default value.

20. The system of claim 15 wherein the central processing unit facilitates determining current values of the one or more parametric controls.

21. The system of claim 15 wherein the central processing unit facilitates determining default values of the one or more parametric controls.

22. The system of claim 15 wherein the central processing unit facilitates determining parametric control capabilities of the one or more parametric controls, including values, value types, and device dependent factory default values.

23. A computer readable medium containing program instructions for:

forming an image processing chain with two or more image processors, the two or more image processors stored in memory, wherein said processors are software modules and each performing a particular type of image transformation, to process digital image data;

providing one or more parametric controls that are uniquely identified and within each of the two or more image processors; and

accessing chosen controls of the one or more parametric controls within each of the two or more image processors to modify the two or more image processors for alteration of the image processing.

24. The system of claim 15 wherein at least one of the parameter controls has a default value that cannot be modified by a user.

25. The system of claim 15 wherein a non-transforming image processor is chained between two transforming image processors.

26. A method for allowing variable controlled alteration of image processing of digital image data in a digital image capture device, the method comprising:

forming an image processing chain with two or more image processors, the two or more image processors being stored in memory, wherein said processors are software modules and each performing a particular type of processing of digital image data;

providing one or more parametric controls that are uniquely identified and within each of the two or more image processors; and

accessing chosen controls of the one or more parametric controls within at least one of the two or more image processors to modify the image processor for alteration of the image processing.

27. A system for allowing variably controlled alteration of image processing of digital image data, the system comprising:

a digital image capture device, the digital image capture device capable of processing digital image data through two or more image processors, the two or more image processors being stored in memory, wherein said processors are software modules and each performing a particular type of image processing and having one or more parametric controls that are uniquely identified; and

a central processing unit within the digital image capture device and capable of linking the two or more image processors to form an image processing chain, wherein the central processing unit facilitates access of chosen controls of the one or more parametric controls within at least one of the two or more processors for modification of the image processor and alteration of the image processing.

28. The system of claim 27 wherein at least one of the parameter controls has a default value that cannot be modified by a user.

29. The system of claim 27 wherein a non-transforming image processor is chained between two transforming image processors.

30. A method of controlling processing of digital image data in a digital image capture device, comprising:

forming an image processing chain with a plurality of image processors, the plurality of image processors being software modules stored in memory located within the digital image capture device, the image processors each performing a particular type of image processing on the digital image data; and

providing one or more parametric controls within each of the two or more image processors, at least one of the parametric controls accessible by a user of the digital image capture device for modifying at least one of the two or more image processors.

31. The method of claim 30 further comprising replacing a parametric control value with a user-specified value.

32. The method of claim 30 further comprising replacing a parametric control value with a default value.

33. The method of claim 30 further comprising exchanging a parametric control setting by an external mechanism.

34. A computer-readable medium having stored thereon instructions which, when executed by a processor, cause the processor to perform the steps of:

forming an image processing chain with a plurality of image processors, the plurality of image processors being software modules stored in memory located within the digital image capture device, the image processors each performing a particular type of image transformation on the digital image data; and

providing one or more parametric controls that are uniquely identified and within each of the two or more image processors, at least one of the parametric controls accessible by a user of the digital image capture device for modifying one of the two or more image processors.

35. The computer-readable medium of claim 34 further comprising replacing a parametric control value with a user-specified value.

36. The computer-readable medium of claim 34 further comprising replacing a parametric control value with a default value.

37. The computer-readable medium of claim 34 further comprising exchanging a parametric control setting by an external mechanism.

38. A system for controlling the processing of digital image data, comprising:

a digital image capture device having an image processing backplane for forming an image processing chain with a plurality of image processors, the plurality of image processors being software modules stored in memory located within the digital image capture device, the image processors each performing a particular type of image processing on the digital image data; and

a parametric control coupled to at least one of the two or more image processors, wherein the parametric control is accessible by a user of the digital capture device for modifying at least one of the two or more image processors.

39. The system of claim 38 wherein at least one parameter control is from a group of parameter controls comprising sharpening control values, color specification control values and compression control values.

40. The system of claim 38 wherein at least one of the parameter controls has a default value that cannot be modified by a user.

41. The system of claim 38 wherein a non-transforming image processor is chained between two transforming image processors.

* * * * *